Multidisciplinary Design and Optimization Framework for Aircraft Box Structures

Reinier van Dijk[a,1], Xiaojia Zhao[a], Haiqiang Wang[a], Frank van Dalen[b]

[a] Flight Performance and Propulsion, Faculty of Aerospace Engineering, Kluyverweg 1, 2629HS Delft, The Netherlands

[b] Fokker Aerostructures B.V., Industrieweg 4, 3351 LB Papendrecht, The Netherlands

[1] Corresponding Author. Tel: +31152782067 Fax: +31152789564 Email: R.E.C.vanDijk@tudelft.nl

Abstract

Competitive aircraft box structures are a perfect compromise between weight and price. The conceptual design process of these structures is a typical Multidisciplinary Design and Optimization effort, normally conducted by human engineers. The iterative nature of MDO turns development into a long and costly process. Knowledge-Based Engineering can be used to automate this process by capturing relevant design process knowledge, which is then re-used inside a computer application. This research will introduce a parametric, generative box model that has been developed using KBE techniques. The generality and rule-basedness of this model allows for the automatic generation of a wide range of box configurations and variants, thereby enabling a thorough exploration of the design space. Structural and price analyses tools have been coupled to the box model to generate the required discipline-specific performance data. With the product model and coupled analysis tools ready, the goal is to automatically optimize for minimum weight and price without human intervention. The design of Gulfstream 650 rudder is considered as initial use case, the first experiences of which are discussed in this paper.

1 Introduction

This paper presents the results of a cooperative research project between Fokker Aerostructures B.V. (Fokker) and Delft University of Technology (DUT). One of Fokker's primary businesses is the design and manufacturing of aircraft box structures (elevator, rudder, aileron and flap) for a wide range of aircraft integrators. The development of such products starts with the proposal phase, during which Fokker conceptually designs an aircraft box structure according to customer requirements, most importantly: a target price and target weight set by the customer. After this process a bid is made. In the near future, Fokker wants to achieve more optimal conceptual box designs during the proposal phase and complete this in less time. To further increase competitiveness, Fokker's desire is to have a software framework that is able to automatically:

1. come up with an optimal concept in 1 working day
2. predict weight within 10% of actual
3. predict recurring price within 10% of actual

To achieve this, DUT has developed a software framework that automates the conceptual design process of aircraft box structures and optimizes these for price and weight. The framework is based on Knowledge-Based Engineering (KBE) and Simulation Workflow Management (SWFM) as the two key enabling technologies. Using KBE techniques, a parametric box structure model has been developed. The KBE model captures and re-uses engineering knowledge to automatically generate the master geometry of the box structure, but also discipline-specific model representations. The latter ability is used to automatically derive three specific representations from the same master model, required to perform FEM, sizing, price and weight analyses. SFWM is used to integrate all software tools into a

single simulation workflow and automate its execution. Optimization algorithms are used to perform iterative MDO studies considering price and weight concurrently.

The paper is structured as follows. Chapter 2 will introduce KBE and SWFM as enablers of MDO. Chapter 3 will elaborate on the generative, parametric box model. Chapters 4, 5 and 6 treat the structural (FEM, sizing), price and weight analysis processes, respectively. Chapters 7, 8 and 9 will introduce the optimization problem more formally, the resulting simulation workflow and some premature results of the overall framework for a Gulfstream 650 rudder component. The paper will conclude with the most important findings and next steps.

2 Multidisciplinary Design and Optimization

Multidisciplinary Design Optimization (MDO) problems typically require numerous design iterations and careful balancing between often conflicting disciplines. MDO takes time, money and needs flexibility. Hence, the need for intelligent automation. Key to the success of MDO is parametric, generative modeling techniques on product side and Process Integration and Design Optimization (PIDO) techniques on simulation process and optimization side. The combination of Knowledge-Based Engineering and Simulation Workflow Management Software offers these capabilities, which will be explained next.

2.1 Knowledge-Based Engineering

Traditional Computed Aided Design (CAD) systems are able to expose product parameters and the external manipulation thereof through an Application Programming Interface (API). However, parameterization is usually limited to a single product topology, making configuration changes hard to perform during MDO studies. Moreover, parameterization is limited to geometrical aspects. Today's complex engineering designs, like aircraft box structures, require the automated analysis of multiple configurations (leading to discrete variables) and multiple disciplines (not only geometry). Knowledge-Based Engineering overcomes the CAD system limitation by capturing relevant engineering knowledge that "teaches" the system how to automatically generate multiple product topologies and inner-topology variations from high-level parameter inputs. Moreover, KBE allows for the capture of other discipline-specific knowledge to derive structural, price and weight aspects. It is not at all limited to the geometry domain.

In this research, the General-purpose Declarative Language (GenDL) was used as the KBE platform [3]. GenDL is a generative application development system for creating web-centric KBE and business applications. The GenDL language is aimed at the general design engineer/programmer, enabling him/her to quickly express the design problem at hand in high-level, object-oriented terms, while still allowing for full access to the functionality of the underlying general-purpose programming language. The GenDL suite is an open platform that blends the power of (ANSI) Common Lisp and NURBS-based geometry kernels, thereby providing both the geometric capabilities of a typical CAD system with the expressiveness/flexibility of a full programming language. With GenDL, an intelligent box model has been developed, which will be discussed further in section 2.3 and chapter 3.

2.2 Simulation Workflow Management

Next to KBE, Simulation Workflow Management software is a key ingredient to MDO. Since MDO processes are typically simulation intensive, involving heterogeneous CAx platforms, SWFM software takes care of workflow automation, including pre-processing, analysis and post-processing steps. Moreover, most of the SWFM systems include various optimization algorithms that allow for optimization studies. In this research, NOESIS Optimus has been used to develop the workflow that integrates the central KBE application, PATRAN, NASTRAN and MS Excel. The optimization results can also be post-processed and visualized in Optimus.

KBE and SFWM are key enablers of MDO, as explained in the foregoing sections. The combination of both technologies allows for a fully automated software framework, known as a Design and Engineering Engine. This will be discussed next.

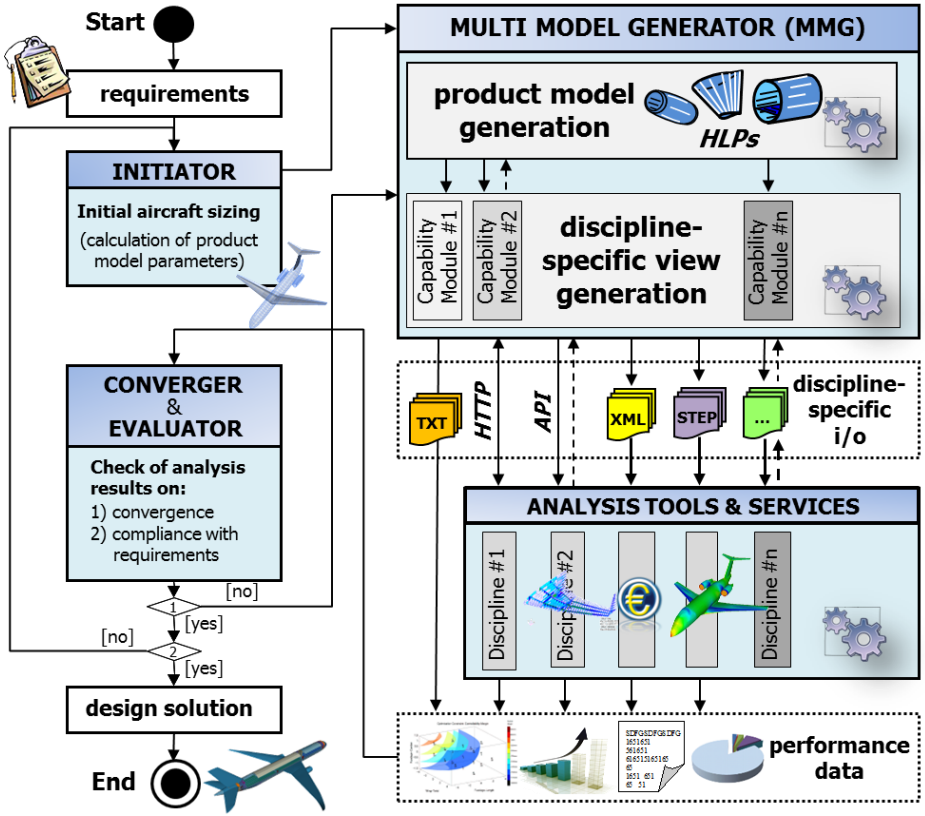

Figure 1: general Design and Engineering Engine (DEE) framework for aircraft. The framework shown is adapted from [1].

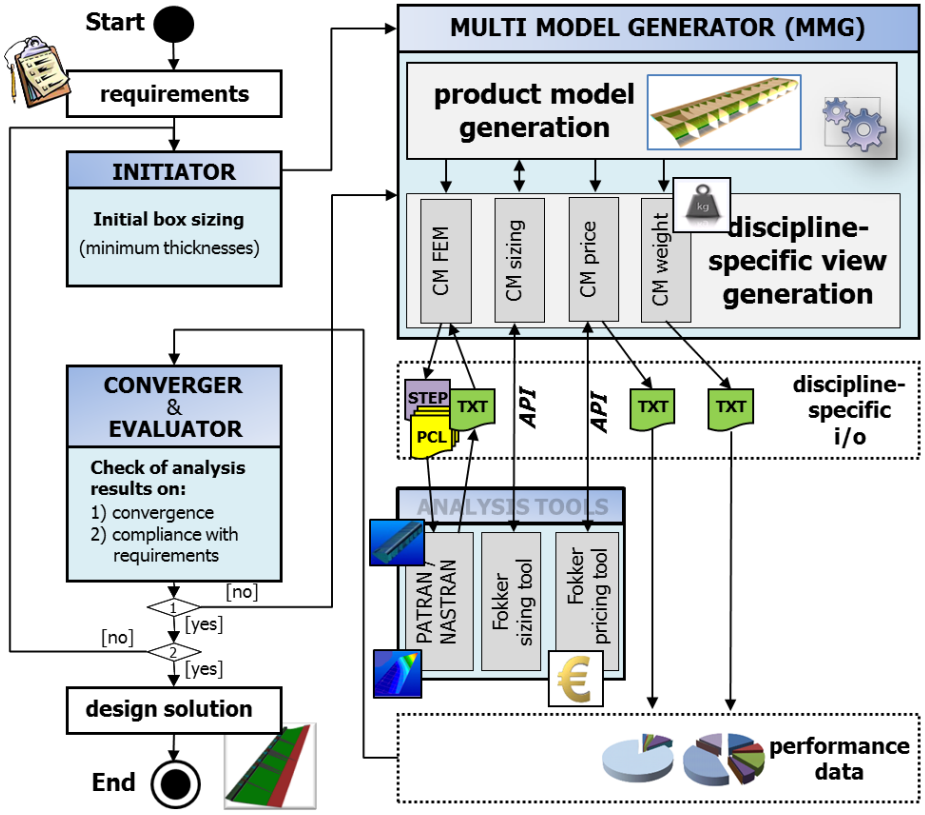

Figure 2: specific Design and Engineering Engine for aircraft box structures. This framework instance has been used in the current research.

2.3 Design and Engineering Engine

Figure 1 shows the Design and Engineering Engine (DEE), a framework conceived for solving MDO problems. It is inspired by the typical system engineering process in which an optimal design is derived from a set of requirements. It translates this philosophy into the software domain. At the heart is a Multi-Model Generator (MMG) component. This KBE application generically describes a product family (hence multi-model) using parametric, generative modeling techniques. Engineering rules are captured inside the program code and take care of automatically generating a product instance from several input parameters, the initial values of which are provided by the user or determined in the so-called INITIATOR component. The INITIATOR can be a DEE framework on its own and captures the product model definition on a lower-fidelity meta-level. It derives an initial parameter set from the top-level requirements that define the design problem. The MMG uses High-Level Primitives (HLPs) as the main building blocks to generate a master (geometry) model. Capability Modules (CMs) are functional, more procedural software artifacts that generate discipline-specific perspectives from this master model. On the basis of these perspectives, CMs may generate input data for external analysis tools or drive these tools through an API or over HTTP in case of web services. CMs can also hold complete functionality to perform a disciplinary analysis autonomously, eliminating the need for external tools. In this constellation, CMs turn into complete "analysis modules". Finally CMs may also update the master model on the basis of analysis outputs. Based on the outputs of the analyses that are considered useful inside an optimization loop ("performance data"), a CONVERGER checks for convergence of the solution. In case no convergence is achieved, a new iteration is set in motion by calling the MMG with slightly updated parameter values. At convergence, the EVALUATOR checks for compliance with requirements. In case of compliance, an optimal feasible solution has been derived. However, without compliance a new design iteration should be initiated with different requirements / constraints. The DEE approach is not limited to the aerospace domain, but applies to the general engineering design field. Van Dijk et al. provide a good example of a DEE implementation in the automotive industry [2].

The DEE instance used in this research is shown in Figure 2. The INITIATOR component is in practice absent. However, it has been kept in place to illustrate the initial sizing strategy of the structure. Chapter 4 will explain that initial thickness values correspond to a minimum thickness laminate. The MMG component uses an HLP building-block approach to model a generic aircraft box structure. Four CMs are developed to extend the MMG with FEM, sizing, pricing and weighing functionalities. These are linked to three external tools, as described in chapters 4, 5 and 6. Finally, Optimus takes on the CONVERGER & EVALUATOR functionality (chapter 8).

Figure 3: Aircraft Box Multi-Model Generator

3 Aircraft Box Multi-Model Generator

The Aircraft Box Multi-Model Generator is a parametric representation of aircraft box structures. This KBE application is at the heart of the DEE (Figure 3). On the basis of the Outer Mold Line (OML) geometry and high-level parameter inputs, this model can automatically generate a wide variety of

inner structure configurations and apply continuous parameter variations within a chosen configuration. Some high-level input parameters for the MMG are given in Table 1. These inputs are mainly related to product configuration or geometry. The different objects in the table can be considered the HLPs of the master (geometric) model, viz. the objects in the software from which any box structure instance is assembled. The configuration related inputs are discrete by nature, e.g. number of hinges, actuators, ribs and spars. Continuous parameters are available to, for example, position structural elements or to describe panel thicknesses. It is important to realize that all inputs may but do not have to be provided by the user, most values are defaulting on the basis of captured engineering knowledge. These values then represent best engineering guesses or are derived from standards. If defaulted values are not appropriate, it is always possible for a user (human or computer agent) to override the inputs.

Table 1: selected set of input parameters for MMG related to geometric configuration

Objects	Aspect	Default	Comment
hinge line	position	*classified*	Two perpendicular offsets from the front spar.
actuator line	position	*classified*	Two perpendicular offsets from the front spar.
hinges	number	4	Minimum of 2.
	position	*rule-based*	A span-wise fraction (0-1) or offset (mm) along the hinge line.
	min. offset	*classified*	The minimum distance between structural elements.
actuators	number	2	Fixed value of 2 actuators for G650 (customer requirement)
	positions	*rule-based*	A span-wise fraction (0-1) or offset (mm) along the actuator line.
	min. offset	*classified*	The offset to hinges and between each other.
spars	number	2	Minimum of 1.
	position, orientation	((0 0) (user user))	Two chord-wise fractions (0-1) at bottom and tip.
main ribs	number	14	Any number is allowed.
	position, orientation	*rule-based* Flight Direction	Combination of span-wise fractions (0-1) and / or angles.
	min/max. pitch	*classified*	The minimum/maximum distance between structural elements.
LE ribs	number	*rule-based*	Any number is allowed.
	position, orientation	*rule-based* Flight Direction	Combination of span-wise fractions (0-1) and / or angles.
	min. offset	*classified*	The minimum offset from hinge-actuator brackets / closure ribs.
	max. pitch	*classified*	Maximum distance between LE ribs (stiffness of LE skins).
splice plates	number	1	Any number is allowed.
	position, orientation	equally spaced Flight Direction	Combination of span-wise fractions (0-1) and / or angles.
	min. pitch	*classified*	The minimum distance between structural elements.
skins, ribs, spars, brackets	thickness	min. t laminate	These values are updated during the sizing step. The maximum attainable value corresponds to the feasibly produced laminate with maximum thickness.
rib flanges	number	*rule-based*	Number of flanges is anywhere between 2 and 3 or 4.
	direction	*rule-based*	Flanges can point inward out outward.
spar flanges	number	2	Fixed value of 2 flanges for G650.
	direction	*rule-based*	Flanges can point in Flight Direction or opposite.

Table 2: selected set of input parameters for MMG related to other disciplines

Objects	Aspect	Default	Comment
hinges	type	*rule-based*	standard, sliding or swiveling.
all structural objects	material	*classified*	Various metallic / composite materials from a materials library with different number of layers, stacking sequences and fiber orientations.
	manuf. method	*rule-based*	Various methods can be selected from a production method library. Certain unfeasible product-production-method combinations are excluded from the choice list in real-time.
connection	assy. method	*rule-based*	
load cases	failure modes	*rule-based*	Describes the full set of failure modes that can happen because of aerodynamic pressure loading, imposed hinge displacements and actuator jamming (and a combination thereof).

Configuration or geometry related objects and parameters are mostly relevant for the master (geometry) model. Various CMs inside the MMG take care of automatically generating discipline-specific models from this master model. However, some these CMs bring new primitives and inputs, of which Table 2 shows a limited overview with aspects from structures and production disciplines. Besides these high-level input parameters, a great deal of model parameters consists of calculated responses based on engineering rules.[1] This may for example apply to flange widths, the number of fasteners required to mechanically join a connection or the rib instance through which hinge loads are transferred. A formal description of the master model and captured engineering knowledge are provided in the next sections. More detail about the CMs is found in chapters 4, 5 and 6.

3.1 Product Model

The aircraft box model is object-oriented. The overall product is decomposed into classes with attributes. Figure 4 shows a high-level UML class diagram with composition ("is part of") and generalization ("inherits from") associations. The top-level assembly class is composed of an Outer Mold Line (OML) that Fokker receives from the customer. The GenDL software is able to import foreign geometry formats. For this work the STEP standard was used to transfer the OML geometry of the G650 rudder. The OML forces inner structural elements to conform to its shape. Constrained by OML dimensions, dedicated input parameters control the position and orientation of one or more spar planes that cut the OML into two or more boxes. The model is assumed to always consist of a Leading Edge (LE) box (either open or closed at the nose) and at least one center box. Several configurations can be achieved. On the basis of two spar planes a 2-spar concept (like the G650 rudder) can be created, in which case the model will also consist of a Trailing Edge (TE) box. Even more spar planes can be used, resulting in a multi-spar configuration with multiple center boxes. All boxes can be further equipped with structural elements. The LE box is composed of two or more hinges and two actuators, each of which can be uniquely positioned along a hinge or actuator line, respectively. Moreover, multiple LE ribs can be placed in the LE box. A center box consists of one or two spars and several ribs. The TE box may have splice plates, dividing the overall box into one or several manufacturable compartments. Spars and rib-like elements (LE ribs, main center box ribs and splice plates) have flanges facing in one direction or facing both directions. The triangular splice-place is usually a double-flanged machined part, resulting in a total of 6 flanges (hence, the cardinalities in the UML diagram). Most of the objects in Figure 4 are further decomposed into other objects, however for simplicity only top-level concepts are shown in class diagram. In practice, the hinge (bracket) is not further detailed to consist of bolts, axle or bearings, because there's typically no knowledge required beyond an empirical total weight estimate and hinge actuation point in the conceptual design phase.

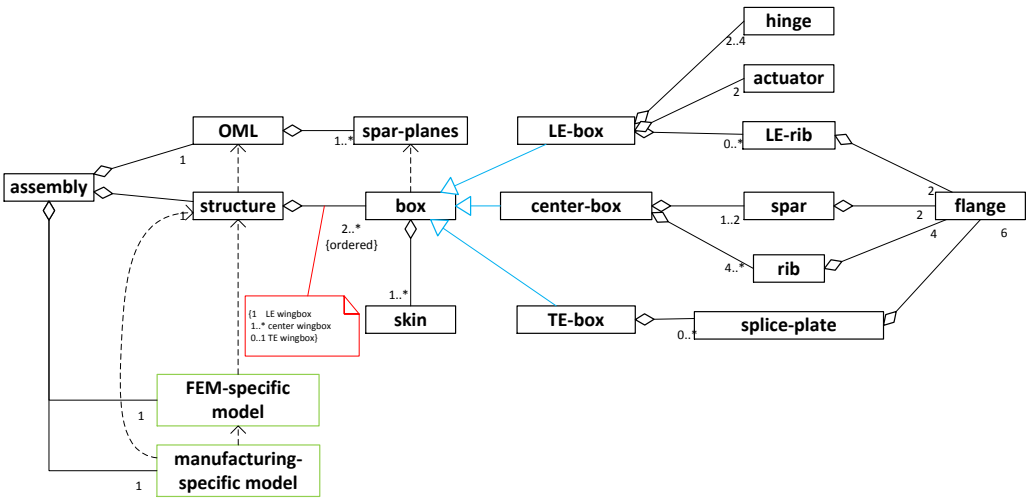

Figure 4: UML class diagram showing the structural breakdown of the aircraft box model

[1] In GenDL-KBE lingo input parameters are called "input slots", while calculated responses are called "computed slots".

Table 3: various examples of engineering rules used inside the MMG for the G650 rudder

ID	Objects	Aspect	Rule[2]	Depends On
A1	actuators	position	Maximize horn arm.	hinge line position
A2	actuators	position	First actuator is located at a fixed offset from middle hinge, second actuator has fixed offset with respect to first actuator.	hinge positions
A3	LE ribs	number position	Two ribs should surround each hinge or hinge-actuator bracket, min/max offsets apply. However, when brackets are close enough to each other, no intermediate rib is required. If the distance between LE ribs is bigger than a maximum pitch X, add extra ribs in between to stiffen the LE skins.	hinges and actuators positions
A4	main ribs	number position	One rib behind each hinge and actuator bracket and one in front of each splice plate to transfer loads.	hinges, actuators and splice plate positions
A5	LE skin	number dimensions	Each LE rib directly facing a hinge-actuator bracket will divide the upper skin in removable panels in order to allow for inspection of the brackets.	LE rib positions
A6	hinges	weight	Empirical formulas are used to calculate weight. A total fixed weight of X kg will be distributed over all brackets according to empirical rules.	N/A
A7	hinges	dimensions	Crack growth analysis requirement: $e/D \geq X$	N/A
A8	flanges	width	Minimum width is X mm for welded joints, for mechanically fastened joints minimum edge distances apply (double row of fasteners in case 3 parts connect).	assembly method connection topology
A9	fasteners	number	At the end of each weld, so-called peel fasteners are installed.	assembly method
A10	fasteners	type	Usually the customer only allows for a limited set of fasteners types, resulting in a limited fastener library.	customer
A11	fasteners	type	The fastener selection rules depend on various aspects, such as: - is the application area on the OML? - is the application area a high-strength area? - is the connection welded? - are connecting member metallic or composite? - etc.	location on structure assembly method customer requirements corrosion protection
A12	ribs, spars	geometry	Make a BREP intersection between the rib reference plane and the OML.	positions OML shape
A13	hinges, actuators	FEM	model the elements as points in the FE model, use RBE2 elements to transfer loads into the rear-facing ribs.	N/A
C1	hinges	number	Minimum number of hinges is 2, in order to meet with fail-safe philosophy.	design philosophy
C2	skins	dimensions	In order to allow for a welding process, the max. skin thickness < X mm.	assembly method
C3	splice plate	position	Don't put splice plates in highly loaded areas.	internal stress
C4	hinges	position	Hinge brackets should have a minimum clearance of X mm between each other.	hinges
C5	main ribs	position	Weld tooling requires rib spacing of > X mm for thin skin areas and > Y mm for thick skin areas.	rib position, assembly method
C6	spar flanges	orientation	Flange-web $\angle$ >90° to make sure product can be removed from mound. Moreover, prevent clash with LE skins.	OML, spar position, assembly method
C7	skins	structural integrity	No skin is allowed to buckle below X% Limit Load (LL), however thin skins are allowed to buckle below 100% LL.	internal stress
C8	ribs, spars, brackets	structural integrity	Stable up to Ultimate Load.	internal stress
C9	LE ribs, ribs, splice plates	position	Align ribs, LE ribs and splice plates in order to make secure joints during assembly. This may also reduce the number of fasteners considerably.	LE ribs, ribs, splice plates
C10	skins, ribs, spars	FEM	All generated FEM surfaces should comply with maximum aspects ratio and skewedness values.	N/A

[2] Values marked X are classified

3.2 Engineering Knowledge

The object-oriented decomposition of the product model is on one hand the formal result of the engineering knowledge underlying the product and process. The result is a generic set of primitives to model the entire design space for the problem under consideration. On the other hand it forms a domain-specific language (or umbrella) for the capture of all other engineering knowledge. Figure 3 provides a semi-formal overview of some important engineering rules for the rudder use case. These rules are mostly expressed in the form of parametric relationships between class attributes, creating dependencies. Their scope might be bound to a single class, however they may also relate attributes of different classes to each other traversing one or multiple association links. All engineering knowledge inside the KBE application will either apply to the aggregation of the product (rules A10 and A11 define which types of fasteners to use) or the values of class attributes. However, it may either manifest itself in a more assertive way or might constrain the feasibility range of certain values. Assertions in this case are factual statements, like the "number of actuators is 2" or "one rib behind each hinge or hinge-actuator bracket" (assertion A4). These knowledge artifacts are used at instantiation-time; they define how objects should be instantiated and what attribute values they will have. Constraints in this case are used after instantiation and are used to check if the resulting product variant is feasible by complying with all requirements, for example "a box structure should always have a minimum number of 2 hinges in order to meet with fail-safe philosophy requirements". It may have happened that a certain product instance might deviate from this through wrong user inputs or automated computations, hence these constraints should be used to check the feasibility of the instance after it has been created. Whenever possible, constraints should be converted into assertive expressions used at instantiation-time. This will assure that only feasible solutions are generated. Constraint C4 was actually converted into a set of assertive expressions that make sure that hinges will never be placed too close to each other. This is sometimes impossible however, in which case constraints are evaluated after instantiation during optimization. Hence, these sources of knowledge will then be captured and re-used inside the optimizer. Product instances that violate constraints, will then be discarded.

The rules in Table 3 are very diverse and may differ by application area or fuzziness. Most rules either influence the number (A3, A9, C1), position (A1-5, C3-5, C9), orientation (C6) or dimensions (A5, A7, C2) of structural members. Other rules define how geometry should be generated (A12). Discipline-specific rules influence either structural (A7, C7, C8), FEM (A13, C3, C10), weight (A6) or production (A8-11, C2, C5, C6, C9) related aspects. The process of knowledge capture and formalization is not straightforward and typically consumes most of the time in KBE application development. Only through iterative use / demonstration of the KBE application and validation of results one can discover if all knowledge has been captured and if the rules are detailed enough. Knowledge formalization is often more complex than initially expected on informal level or made impossible by lack of preciseness. For example, A1 states that the actuator line should be positioned such that a maximum horn arm results. To guarantee this, an extensive set of geometrical operations was required to position this line as close as possible to the LE skin. The implementation of LE rib positioning conform A3 resulted in a 100 lines of code algorithm with many conditionals. Constraint C6 was not precise enough and needed another round of detailing, hence answering "when is the angle exactly big enough?". Or the logic underlying a rule might be too fuzzy. It was impossible to accurately convert C9 to an assertive expression: what component takes on the leading role in rib alignment? Does the main rib influence the positions of LE ribs and splice plates, or is it the other way around? Sometimes rules may not end up inside the KBE application, but for example in the optimizer. C1 defines the minimum bound of a variable in the optimization problem, while C6 and C7 define constraints that the optimizer should check. Not satisfying this, results in infeasible structures.

Rules are enablers of the KBE approach, however they also significantly limit the validity of the application. For example constraint C7 (Limit Load requirements) renders the application only valid for post-buckled concepts like the G650. Constraint C9 (rib alignment) is specialized to mechanical fastening, the alignment of structural members may be a bad choice in case of welding.

Assertion A6 defines the strategy of how to assess hinge (-actuator) bracket weights through empirical rules. This may limit the accuracy of results and might require more sophisticated modeling strategies for bottom-up calculations. One has to keep in mind that the model is as good or valid as the rules that went inside. Traceability of engineering knowledge that goes inside a KBE application is of utmost importance in this respect.

Finally, KBE application scope should be clearly agreed upon before implementation. Fastener selection rules (A11) turned out to be so exhaustive that it would require extra extensive interviews and knowledge capture sessions, while the benefits are not immediately clear. As a result, it was decided to use a single averaged fastener type in the conceptual design phase.

4 Structural Analysis and Sizing

This section presents the structural analysis module that was developed in order to facilitate the weight-price trade-off. Due to its tight coupling with the MMG, designers can directly obtain performance results from high-level inputs and relate these back to the various design options they explore in a conceptual design phase.

4.1 Strategy

It has been the goal of this research to increase the level of fidelity of structural analysis in the proposal phase to increase the accuracy of weight estimates. For this reason, the MMG has been coupled to a FEM solver (NASTRAN). The creation of the FEM-specific model has been fully automated, by capturing relevant domain knowledge that underlies these activities. While the actual meshing of the geometry is "delegated" to the standard pre-processor of the FEM solver (PATRAN), rules are used to subdivide the overall geometry into easily meshable surfaces, thereby assuring congruency constraints. From computational performance standpoint, the sizing strategy was decided to follow a single-step approach. In this approach, all surface thicknesses are initially set at their potential minimum; each value equal to the thinnest laminate that can be manufactured. On the basis of this minimum-thickness model, running loads are derived in the FEM solver. These results are then transferred to a Fokker in-house developed sizing tool. This tool can determine the most appropriate laminate from a laminate library that can sustain the respective loading pattern on the basis of panel dimensions and running loads. The single-step approach is never 100% accurate as updated thicknesses / layers will influence the structural behavior and ideally several iterations are performed. However, Fokker experience dictates that differences are not significantly big and this strategy is adequate for proposal purposes. This compromise obviously greatly benefits the computational performance.

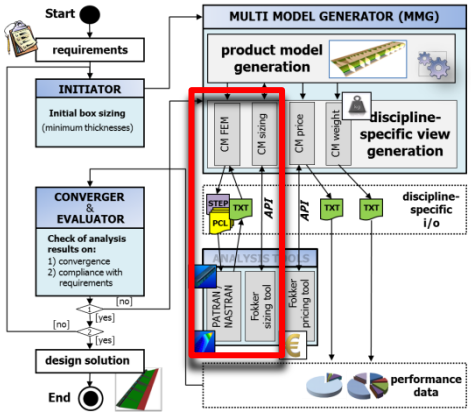

Figure 5: structural analysis process in DEE

4.2 Implementation

The structural analysis module is spread over two branches of the DEE, as shown in Figure 5. Moreover, a detailed activity diagram is shows in the "FEM" and "sizing" partitions of Figure 6. Four software components are involved: a CM for FEM pre-processing, PATRAN/NASTRAN as FEM pre-processor/solver, a CM for surface sizing wrapping around an Fokker in-house developed sizing tool.

4.2.1 Capability Module: FEM pre-processing

The CM for FEM pre-processing transforms the master geometry into a FEM-specific perspective with appropriate geometry and object properties. This perspective conveniently allows for the automatic generation of PATRAN input files that contain model pre-processing instructions and geometric data. The surface subdivision unit extracts all surfaces from the master geometry and splits them

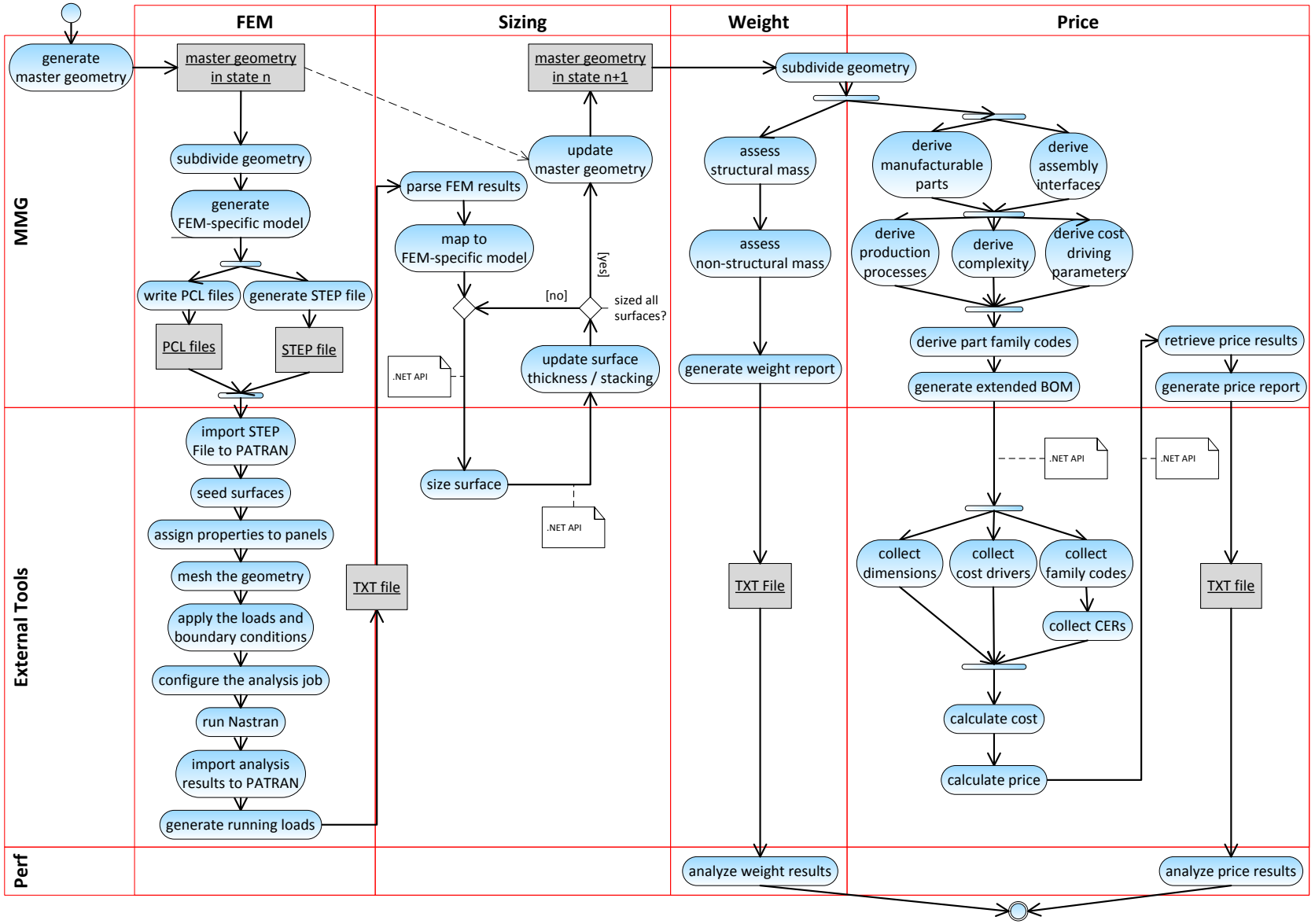

Figure 6: UML Activity Diagram with two-dimensional partitioning. Horizontal swimlanes relate to software components, vertical swimlanes to engineering disciplines

into smaller ready-to-mesh surfaces. This transformation assures surface congruency and satisfies aspect ratio / skewedness constraints. The process is visually depicted in the left branch of Figure 7. The subdivision unit first separates the entire OML into boxes by intersection with all spar planes. It orders these from LE to TE. Next, all rib-like parts are collected (LE ribs, main ribs and splice plates). Although ribs may only exist in a single box, all their reference planes are extended over the entire chord of the OML. In case ribs connect over different boxes, their reference planes are merged. Each reference plane is extended in a direction that will assure maximum separation. For each box the ribs planes are collected that intersect the box geometry and an assessment is made whether the plane is physical (present in box) or virtual (resulting from the extension). Next, each box is recursively separated into smaller solids (segments) by intersection with the rib planes. Finally, all solid faces are collected into ordered lists. This collection of faces should only include physical faces and therefore follows two filtering steps. First, all duplicate faces are removed in both chord-wise and span-side directions. Second, top (tip) and bottom (root) faces of a segment are removed in case they were the result of intersection with a virtual rib. The final collection of surfaces is ordered and a unique ID is assigned to each surface for traceability. At this stage the master model has been transformed into meshable FEM geometry and all surfaces are written into a single STEP file ready to be imported by PATRAN.

Figure 7: two subdivision steps are required to derive the appropriate perspective for both FEM analysis (left) and manufacturing / weight analyses (right). The red boundaries highlight the domain-specific segmentation.

Each surface in the FEM-specific model is not limited to geometrical aspects only, an associative link with the product model is kept intact. Hence, each surfaces holds non-geometrical properties relevant for analysis, e.g. material properties. Next, all non-surface like elements are collected, namely hinges and actuators. Also these objects are ordered and uniquely numbered for traceability. Moreover, an association is made between the hinge/actuator and the rear-facing rib surface into which the introduced loads should be transferred.

The final step in the FEM pre-processor is the creation of a set of PATRAN instructions, using the PATRAN Command Language (PCL). This part of the software is named "PCL-writer" and contains a general set of utilities that can generate PCL code from high-level inputs. The PCL-writer makes full use of the KBE system language to assess all information available in the FEM-specific

model. The PCL-writer is a means to automate the PATRAN model set-up and meshing steps (Figure 8), but is also the means to re-assign the properties of finite elements inside PATRAN that would normally go lost when only a STEP file is imported. The PCL-writer manifests itself in the creation of several PCL files. The PCL writer component is generic enough for any GenDL-developed KBE application to seamlessly link with PATRAN/NASTRAN. Wang et al. [10] re-used it for the structural analysis of aircraft fuselage panels.

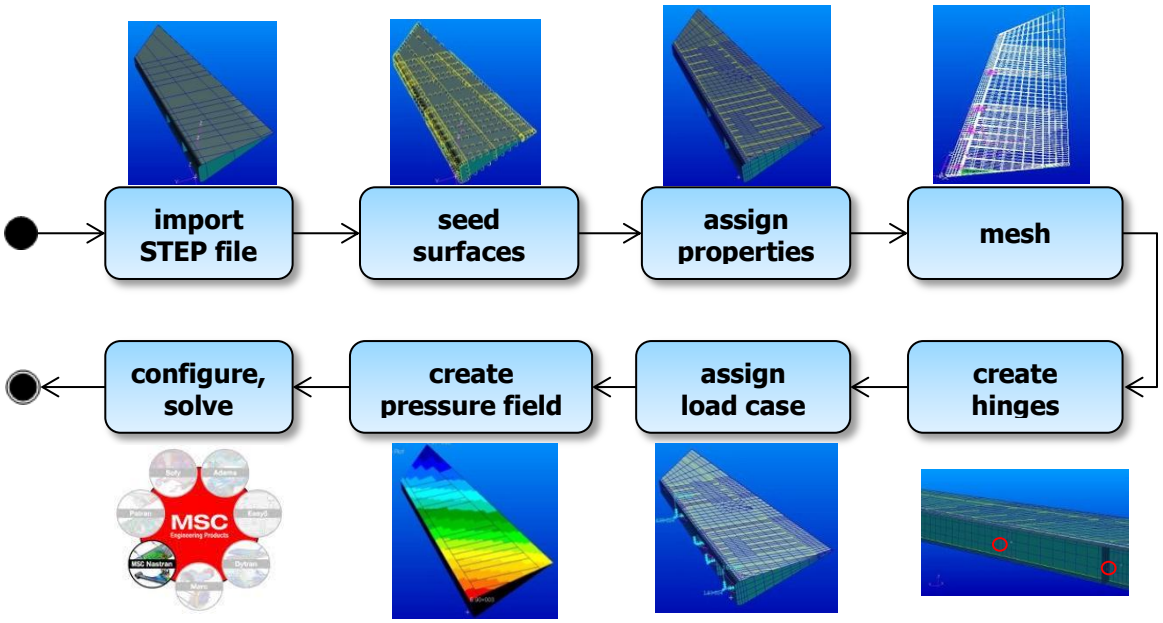

Figure 8: an execution example of the automated structural analysis

4.2.2 Model Generation and Execution

The rudder FE model is generated in PATRAN and analyzed in NASTRAN. The skin plates and spar/rib webs are modeled as CQUAD elements, and the spar caps and rib flanges are modeled as CBAR elements. RBE2 elements are used to simulate the load transfer from hinges and actuators into the connected ribs. A pressure field is applied to the rudder skin to simulate the aerodynamic pressure and an enforced displacement is applied to hinges to model the loads due to vertical tail bending. The solver calculates the running loads for each surface for a multiple of load cases as a result of aerodynamic pressure loading, imposed hinge displacements and actuator jamming (and a combination thereof). In a manual human-driven modeling process, the following activities are identified as repetitive work when making FE models:

- importing geometry
- assigning properties to the geometric entities
- meshing the geometry
- applying the loads and boundary conditions
- configuring the analysis problem
- submitting it to the analysis solver
- post-processing the analysis results

All these activities are completely automated through the PCL file medium. Figure 8 shows the automation process for the G650 rudder FE model. The output consists of all running loads around the different skin surfaces, spar and rib webs.

4.2.3 Capability Module: sizing

The PATRAN results file generated through PCL automation is retrieved and parsed by the sizing CM inside GenDL. This module maps running load results back to all surfaces in the FEM-specific model

by matching of IDs. Next, an iterative connection is made to the Fokker in-house developed sizing tool. To this end, a bi-directional Excel link is established between GenDL and Excel through .NET libraries using the RDNZL package for Common Lisp [13]. For each surface, running loads and dimensions are automatically written into the Excel sheet and the best laminate is retrieved from the tool. All laminate properties like thickness, number of layers, stacking sequence and fiber orientations are then mapped back onto the surface instance inside GenDL. While iterating over all surfaces, the same software link is kept alive (and not continuously re-established), greatly improving computing performance. Finally, the thickness results of all individual surfaces need to be mapped back to the original master geometry model. The idealized thicknesses on surface level need to be translated to the higher-level physical model. Therefore, the CM will update the original minimum thickness model (instance *n* in Figure 6) in the master geometry with new thickness values (instance n+1 in Figure 6) for all parts. The algorithm first adds thickness to the areas where strong ply drop-off rates are encountered, and finally calculates a single volume-averaged thickness. For the first software prototype this approach is deemed adequate. However, a next version might optimize all stacking sequences to assure consistent interleaving solution and hence a minimum addition of weight. The volume-averaged thicknesses are the starting-point for price and weight analyses.

4.2.4 Sizing Tool

A Fokker in-house developed sizing tool was used for each surface to find a laminate with minimum thickness and associated stacking sequence that can sustain local failure modes, such as buckling and static failure. The inputs for the tool are surface dimensions, compressive and shear loads. All laminates in the laminate library have a fixed stacking sequence of $[0\ \pm45\ 90]_s$, which is common practice for box structure conceptual design. The sizing tool was validated by the NASTRAN 105 solver and is considered as a black box for sizing purposes.

5 Price Analysis

This section presents the cost and price analysis module that was developed in order to facilitate the weight-price trade-off. The analysis has been developed particularly for aircraft box structures and was embedded in the DEE framework. Due to its tight coupling with the MMG, designers can directly obtain cost-price information from high-level inputs and relate the results to the various design options they explore in a conceptual design phase. Moreover, KBE techniques, such as knowledge extraction and analysis process automation, are intensively used for the price analysis, which largely reduces the response time compared to traditional excel-based price estimations and establishes a consistent and maintainable tool. Figure 9 shows the positioning of this module in the DEE.

5.1 Strategy

Typically, cost estimation methods are categorized into two classes: top-down and bottom-up. The top-down method delivers an overall estimate of a whole product, which then can be broken down into its sub-components according to either the product structure or an associated cost breakdown structure. Rand corporation developed an airframe acquisition cost model based on the parametric relations associated with aircraft unit weight and speed in the mid 70's [6]. Raymer incorporated some modifications on the Development and Procurement Costs of Aircraft (DAPCA) IV model, although high level parameters were still utilized for the cost estimation [7]. The bottom-up method establishes detailed estimates based on individual components or manufacturing processes, which can be then accumulated to provide the full estimate. Other work of interest includes that of Hanffner who developed cost models for composite materials w.r.t. manufacturing processes in 2002 [8] and Van der Laan implemented a process-based cost estimation for aircraft movables [9].

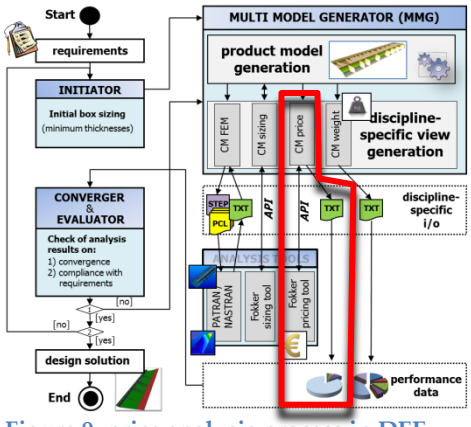

Figure 9: price analysis process in DEE

Top-down estimates are limited in that they cannot provide a detailed insight of a product cost, whereas, bottom-up estimates can be incomplete as it is difficult to obtain all the required inputs at the same level of fidelity from the conceptual design phase. As a consequence, a cost estimation facility has been developed that is a compromise between top-down and bottom-up methods. It is able to provide an estimate with greater accuracy than a top-down method and on the basis of a smaller amount of parameters than would usually be required in a bottom-up approach. The estimation facility follows a part family philosophy, in which manufacturable parts and assembly interfaces are grouped in families according to their main characteristics. For each part family, Cost Estimation Relationships (CERs) have been developed using statistical and empirical relations. Driving parameters such as length, area, weight, part number are included in the corresponding part family CERs, as well as other reference values. The overall price estimate is then based on associating CERs to parts on the basis of their part family code and accumulating results. Of course, the entire set of part families should be general enough to handle all variations in the box structure model. In this research, a total of 258 part families have been derived from characteristics like size, shape complexity, material, manufacturing method and in-house development / outsourcing. In addition, KBE techniques were employed to capture all knowledge to automatically derive the aforementioned property values from the geometric master model and high-level user inputs.

The price estimate is built up from recurring and non-recurring cost and incorporates extra (profit) margins. A high-level breakdown is shown in figure. 2. In this research, non-recurring aspects are considered out of scope. The recurring cost items that are most dominant and are grouped into material and labor cost (thereby ignoring cost associated to support, inspection, etc.). The material cost is derived from material rate, part length, width, height, density and chipped rate, as shown in equation 1.1. The labor cost is estimated based on manufacturing hours and labor rate, where the manufacturing hours is derived from the driving parameters that help define the part family, see equation 1.2. The rates for both material and labor cost vary over the different part families.

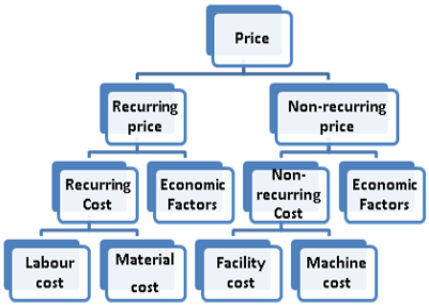

Figure 10 high-level price breakdown

$$C_{material} = r_{material} wlh\rho n(1+r_{chipped}) \tag{1.1}$$

$$C_{labour} = r_{labour} \cdot h(x) \tag{1.2}$$

The price estimate is then a summation of recurring cost and includes an extra margin that is driven by economic factors such as exchange rate, profit, surcharge rate, learning curve and General and Administrative (G&A) expenses. In this research, a financial rate f based on factorization of the recurring cost was defined to synthesize the aforementioned economic factors.

$$P = \prod_{i=1}^{n} r_i \cdot C = f \cdot C \tag{1.3}$$

5.2 Implementation

The price analysis module is a single branch of the DEE, as shown in Figure 9. Moreover, a detailed activity diagram is shown in the "Price" partition of Figure 6. Two main software components are involved: a CM for price analysis pre-processing and a Fokker in-house developed pricing sheet. The MMG provides the master geometry that the CM pre-processes to enable an analysis. An extended Bill Of Materials (BOM) is generated as input for the actual price calculation. The price is then calculated within the analysis module and a final price report is generated.

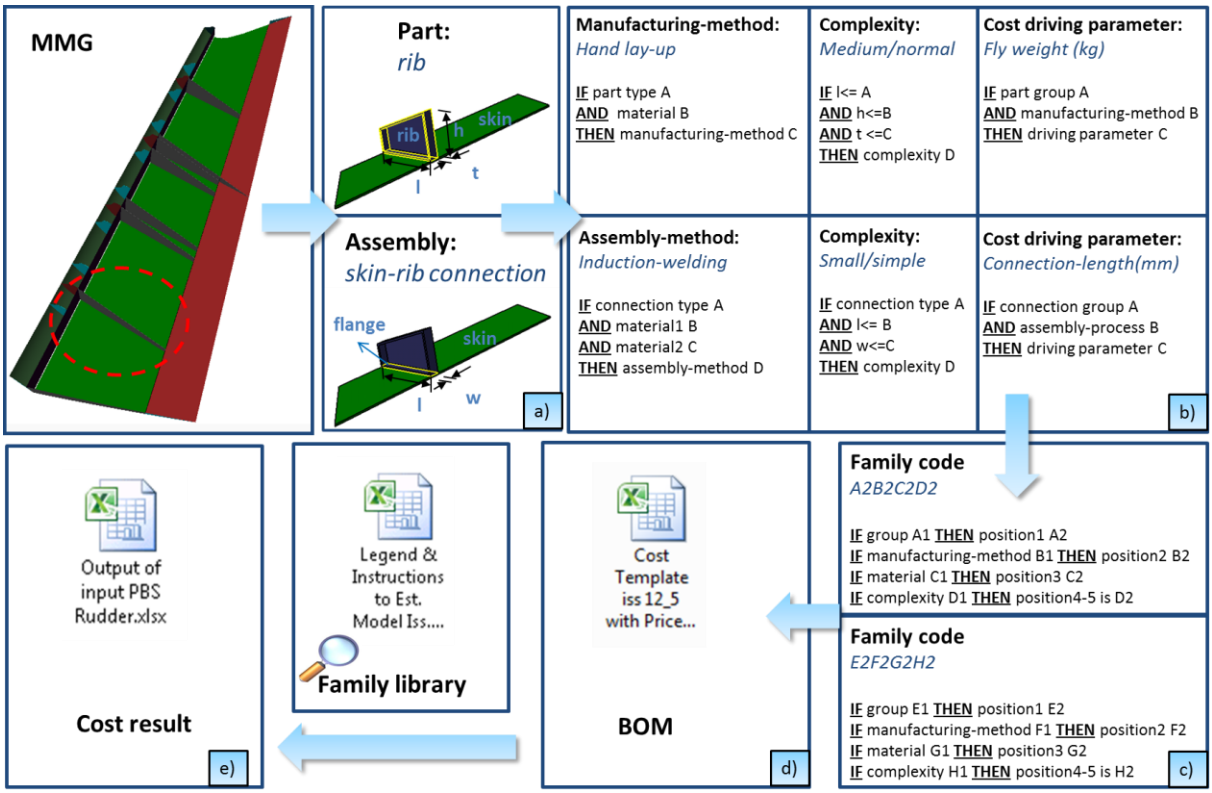

Figure 11: execution examples of rib part and rib-skin connection

5.2.1 Capability Module: manufacturing pre-processing

This CM integrates the parametric box model with the price analysis module. To support the analysis, a manufacturing-specific model is required. Moreover, this CM is implemented in GenDL [3], while the analysis module is an Fokker in-house developed Microsoft Excel workbook. Hence, a data transformation is required that transforms the manufacturing-specific model properties into an Excel worksheet. A total of four sequential pre-processing steps implement this transformation.

1. subdivide geometry
   The manufacturing perspective should include manufacturable parts and all assembly steps (Figure 11a). To this end, manufacturing/maintenance rules have been captured to derive this perspective automatically. Figure 7 (right panel) shows that for the G650 configuration, the TE box is divided into two skin panels. This knowledge is inferred from dimension constraints on skin parts. Moreover, the fixed LE skin on left-hand side actually consists of several panels, due to accessibility requirements to the hinge-actuator brackets. Besides parts, all connections between parts are derived. Connection types include skin-rib, skin-spar, skin bracket, spar-rib and rib-rib.
2. derive properties
   The individual part and connection properties, such as manufacturing/assembly process type, shape complexity and cost driving parameter, are also derived from captured rules. For instance, the manufacturing process is derived from the combination of part type (skin, rib, spar, and bracket) and material. Similarly, the assembly process is derived from the connection type and the combination of materials of each part in the connection. Figure 11b illustrates this rule-based inference mechanism.
3. assign family code
   In order to uniquely characterize a part or connection from a costing perspective, the properties derived in the previous steps are condensed into an alphanumeric family code. Those symbolic codes are used inside the analysis module to associate the part with a CER from the part family library. The rules to transform general part properties into a part family code have been captured inside the CM (Figure 11c).

4. generate extended BOM
   The previous steps create a set of knowledgeable parts and connections for each product topology. This final step transforms all information into an Excel worksheet that represents the "extended" BOM (BOM + cost driving parameters) and serves as input for the Fokker analysis tool. The same Excel interface as described in section 4.2.3 was used for automation (Figure 11d).

5.2.2 Analysis Module

Once the extended BOM is written to the Excel environment, product cost and price are automatically determined. For each BOM entry the part family code relates a part to CERs for material and labor cost in the part family library. As input to the CERs, properties such as dimensions and cost driving parameters are also extracted from the BOM. Finally, a cost aggregation is carried out for the entire BOM and a total price estimate is generated that includes both cost results and economic factors.

5.2.3 Results Pre-Processing

The total price results are summarized on a dedicated worksheet in the Excel workbook. It doesn't only categorize results into recurring material and labor price, but also logically distributes results over different family codes and production processes. Cost driving parameters are also shown, providing insight in the most important sources of cost. In this way, designers will be aware of the impact of their choices and adopt a design-for-cost philosophy during the proposal phase. To ease the interaction with the optimizer and keep a single source of truth, the bi-directional GenDL-Excel interface is used to extract the key price figures from the workbook (Figure 11e) and store everything centrally in the product model. From there, a simple computer-readable TXT file is generated that contains only price responses relevant for the optimization study.

6 Weight Analysis

The weight analysis module is a single branch of the DEE, as shown in Figure 12. A detailed activity diagram is shown in the "Weight" partition of Figure 6. Only one software component is involved, namely a CM for weight analysis. This CM depends on the geometry subdivision unit introduced in section 4.2.1. The complete analysis code was written inside GenDL turning this CM into an "analysis module" inside the MMG. A "Class 2.5"[3] weight estimation method has been developed based on the updated thickness model and extra empirical formulas. All areas of skin panels, spar and rib webs are computed in GenDL as well as the length of spar caps and rib flanges. The mass is then easily calculated by multiplying the material density and surface area with thickness values from the manufacturing-specific model. Since numbers of fasteners and fastener types are also explicitly modeled, their mass is calculated bottom-up. For sealing and painting masses, average thickness assumptions are made. In combination with exterior surfaces areas, the mass values can be assessed. Only the non-structural masses for hinges brackets, hinge-actuator brackets are empirically assessed. At this stage, all weight results are now also present in the product model, creating a single source of truth for all design information. A simple computer-readable TXT file is generated that contains only weight responses relevant for the optimization study.

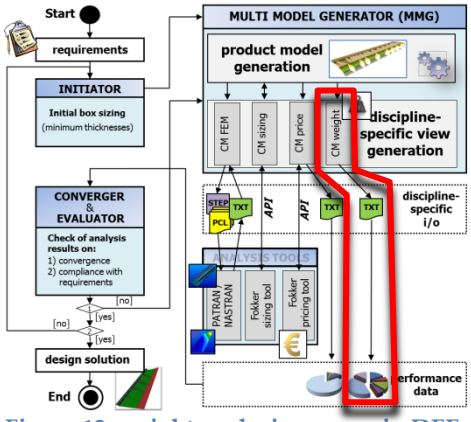

Figure 12: weight analysis process in DEE

[3] Most of the weight assessment is done on Class III [5] bottom-up basis, however for hinge and actuator weights, Class II [5] empirical formulas have been employed, rendering the overall weight assessment method "Class 2.5".

The chain of operations described in chapters 3, 4, 5 and 6 is fully automated. Hence, when product variations occur inside the MMG, analysis models are updated and price-weight calculations will be re-executed. This ability to iterate enables optimization studies.

7 Optimization Problem

The objective of the optimization is to optimize the rudder design to reduce cost (maximize price) and weight. The product focus is on skin panels (with varying thickness), ribs and spars and hinges/actuators (as black boxes). The optimization problem deals with the overall topology ("master geometry") and the sizing of individual parts. Ideally, the objective function to be minimized in the optimization is cost, while structural strength and weight are constraints. In case the weight and strength constraints cannot both be satisfied, the objective function becomes a weighted function of weight and price, while the strength constraint must in every case be satisfied.

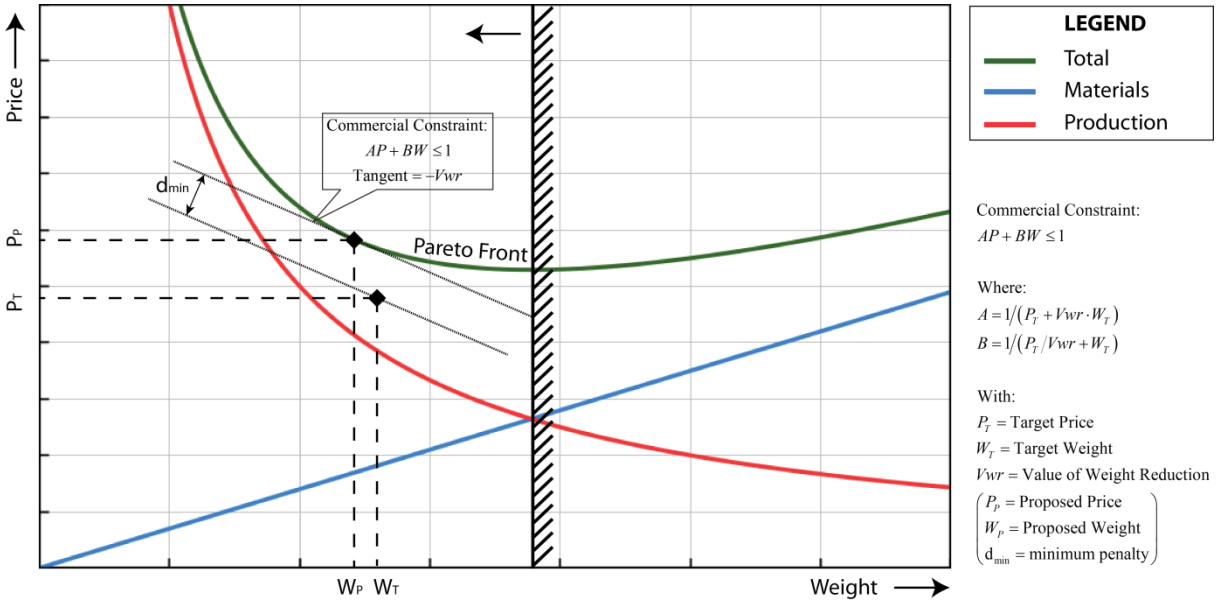

Figure 13: graphical representation of the optimization problem

Figure 13 illustrates that the commercial constraint generally cannot be satisfied due to the challenging levels of Target Price and Target Weight. In such cases, Cost is abandoned as the optimization objective, and the Commercial Constraint becomes the objective function to be minimized instead. The resulting problem formulation is given in equation 7.1.

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & J(\mathbf{x}) = A \cdot \text{Price}\left(\mathbf{x}_1^T, \mathbf{x}_2^T\right) + B \cdot \text{Weight}\left(\mathbf{x}_1^T, \mathbf{x}_3^T\right), \quad \mathbf{x} = \left(\mathbf{x}_1^T, \mathbf{x}_2^T, \mathbf{x}_3^T\right) \\ \text{subject to:} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p \end{aligned} \tag{7.1}$$

Hard constraints which must always be satisfied are related to the structural integrity of the product:

$$\text{Reserve Factor} \leq 1 \tag{7.2}$$

Price will be estimated using, as a basis, the existing Fokker cost sheet. Weight is assessed using a Class 2.5 estimation method. Reserve Factors are calculated for each finite element, every load case and every failure mode. Failure modes include material strength, damage tolerance, bearing-bypass, open hole tension etc. Local buckling and Euler buckling are not evaluated at element level, but at the level of a web, skin pocket, super stringer or complete structural component using the Fokker sizing

sheet. The major design question that the optimizer should be able to answer from a price/weight/performance perspective is: "Is it more valuable to add/remove single ribs or increase/decrease skin pocket thickness?". Table 4, Table 5 and Table 6 give an overview of the most important parameters, variables and constraints in the optimization.

Table 4: fixed parameters

Name	value[4]	Unit
No. of spars	2	-
Initial panel thicknesses	X	mm
Rib material	T300/PPS, ply thickness = X	-
Spar material	T300/PPS, ply thickness = X	-
Skin material	T300/PPS, ply thickness = X	-
Actuator positions	X mm offset from root	mm
Hinge line offsets	X	mm
Actuator line offsets	X	mm
Total hinge mass	X	kg
Non-structural mass factors	X	-

Table 5: optimization variables

Priority[5]	Name	Type	Range	Unit
1	A (weight factor price)	Continuous	X	-
1	B (weight factor weight)	Continuous	X	-
1	No. skin panel plies	Discrete	Min: X max: X	-
2	Ply layup method	Discrete	Hand layup / ATL / AFP	-
2	Ply width (ATL/AFP only)	Continuous	X mm – X mm	mm
1	Laminate library	Discrete	Set of n variables, i.e. fiber orientations for each next ply added to the laminate	degrees
1	No. of center box ribs	Discrete	Range from X – X, step 1	-
2	No. of hinges	Discrete	2-5, step 1	-
1	LE rib pitch	Continuous	Range from X-X	mm
1	Rib thickness	Discrete	(X-X)	mm
1	Spar thickness	Discrete	(X-X)	mm
1	Rib positions	Continuous	Root to tip of rudder (0-1)	-
1	Rib orientation	Continuous	Range from X-X, step X (angle)	degrees
1	Spar positions	Continuous	(0-X) fraction of chord	-
2	Hinge positions	Continuous	Root to tip of rudder (0-1)	-
2	Hinge type: suppressed Degrees Of Freedom (DOF)	Discrete	For each hinge type, 6 digits having values 0 (free) or 1 (suppressed) indicate DOFs which are suppressed.	-
1	Load case	Discrete	Set of load cases for Air Loads plus imposed deformations on hinge line to represent Fin bending.	-
1	Structural Configuration	Discrete	Multi-rib	-
2	Structural Configuration	Discrete	Multi-spar Sandwich	
1	Local buckling onset	Continuous	Function of laminate thickness (% LL)	-
2	Spar and Rib flange orientation	Discrete	Fwd./Aft, Up/Down	-
2	Type of part tooling	Discrete	Male (Inner Mold Line, IML) or Female (Outer Mold Line, OML)	-
1	Joint type	Discrete	Bolted / Induction Welded	-
2	Joint type	Discrete	Single Step Assembly/ Welded/ Butt-jointed/Bonded/Co-cured	-
2	Joint shimming	Discrete	Yes / No	-

[4] Values marked X are classified

[5] The first version of the framework includes items with priority "1", and the architecture should support the later implementation of items with priority "2".

Table 6: optimization constraints

Priority	Name	Expression	Unit
1	Target weight / price	A * Weight + B * Recurring Price ≤ 1	-
1	Reserve factors	RF ≥ 1 (Note: applies to each load case and each failure mode.	-
1	No. of skin plies	n > n_min	-
1	panel thickness	constrained by manufacturing method	mm
1	Laminate Library	Local build-up of ply orientation according to Laminate Library	-
1	Ply drop-off rates	$\Delta t / \Delta x < X$ $\Delta t / \Delta y < X$	-
2	Tape layer constraints	Ply contour shape to be manufacturable by specified tape laying machine (if applicable)	-
2	OML Tolerance	Deviation of outer aerodynamic surface contour w.r.t. nominal	mm

8 Optimization Workflow

In order to perform the actual MDO study, a simulation workflow is under development. NOESIS Optimus has been selected as SWFM software. The current version of the OPTIMUS workflow is shown in Figure 15. For convenience, it has been drafted in a way that closely resembles the theoretical DEE schematic (Figure 2). Red dashed frames highlight different fundamental process aspects. The overall workflow integrates three GenDL modules: the Aircraft Box MMG (*MMG*) and capability modules that are responsible to create the appropriate perspectives for FEM and manufacturing disciplines (*CM1, CM2*). The workflow drives PATRAN/NASTRAN for automatic stress analysis (*A1*) and two other custom-made Fokker/GenDL applications for price/weight estimates, respectively (*A23*). All GenDL applications are set up as web services, which means they run on a remote server with internet access. This gives rise to a Service-Oriented Architecture, where communication follows the HTTP REST protocol. File exchanges between GenDL and PATRAN occur twice and use the SFTP protocol. The workflow starts with an array of design variables, a limited overview of which is given in Figure 14. Only for the first iteration of the MDO study, a new rudder instance is created in the MMG with appropriate inputs. Subsequent iterations update that instance in real-time by changing parameter values. For each instance, a feasibility check (*F1*) is performed to check if a proper model was created. Since a great majority of the involved constraints are geometric in nature, it was a natural choice to evaluate these constraints inside the MMG and let GenDL communicate a single flag value back to Optimus. If the model is feasible at this stage (flag = 1), the workflow continues with a request to the structural capability module (*CM1*) to derive the FEM perspective and generate the related CAD geometry and PCL scripts. It transfers these to PATRAN, after which PATRAN is executed (model set-up), NASTRAN is executed (solver) and PATRAN is once more executed to generate an output report with stress results per FEM surface (*A1*). The results are then transferred back to the MMG, which uses these to perform a one-step sizing method (chapter 4) and update all surface thicknesses and material compositions. At this stage, a second check is performed to see if the rudder instance is structurally feasible (*F2*). Only if for each surface an adequate material was found from the laminate library and all reserve factors are met, or in other words if the rudder is structurally sound, the workflow continues. In this case a manufacturing perspective (*CM2*) is derived (chapter 5) and customized modules automatically derive price and weight estimates (*A23*) from this perspective. The final report files contain all important performance data from which the output variables are bound. In case these outputs do not yield an optimal solution, a new iteration is started with a different set of design variable values.

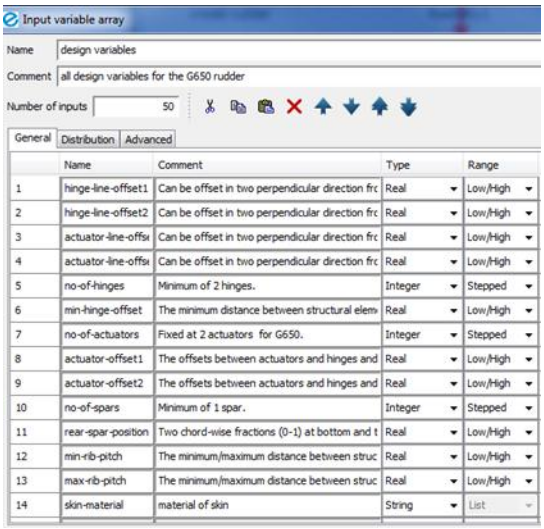

Figure 14: limited list of design variables

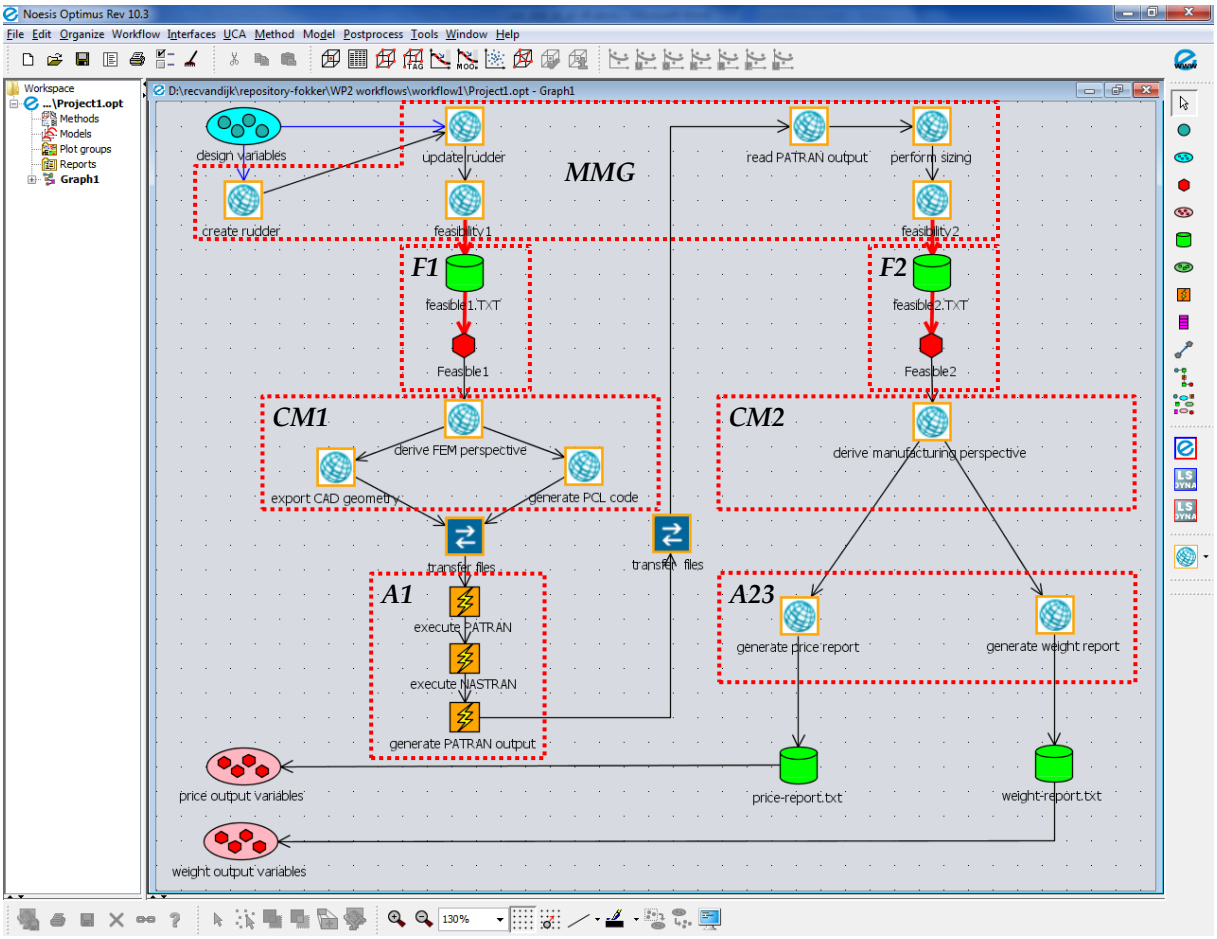

Figure 15: Optimus workflow developed for this research.

9 Results

At this stage of the research, there are no MDO results yet. The delivery of the framework and description of it, are therefore the major results in itself. However, with the development process in hindsight, two important results can be derived from a business point and code complexity point of view.

9.1 Business Case

Verhagen et. al [14] show that former KBE research rarely quantifies the savings resulting from a KBE-enabled process. Moreover, development costs are never shown. This section will therefore present a quantitative KBE business case for this research by comparing the expecting savings and actual software development investments. The authors challenge future KBE publications to use a similar concrete approach. In this way, literature can also clearly proof industrial relevance.

The goal of this research has been to develop a software framework that can automatically design and optimize an aircraft box structure for minimum price and weight in the concept proposal phase. The main benefit of the resulting level of automation lies in a reduction of engineering time spent on repetitive work. In a human workflow, activities that normally qualify as repetitive work are CAD model set-up, discipline-specific model creation (FEM, price, weight) and simulation execution. Extra time is usually also spent on the search for, transformation of and distribution of data and information. This software framework completely eliminates the needs for such wasteful activities through workflow integration. The interaction between the new software framework and human engineers, results in a shift of responsibilities. While the software framework now takes over the very procedural tasks, the new role of design engineers is to be creative and come up with best guesses of

initial model inputs, definition of problem objectives and constraints, to interpret final results and respond with different design inputs.

The software framework affects the bottom-line by reducing the engineering hours required for proposals. This reduction has a direct value in cost savings. Moreover, Fokker Aerostructures can use the extra available time resulting from automation in two ways:

1. reduce lead time and take on extra projects, thereby increasing capacity.
2. perform more design iterations, resulting in higher quality end products.

The extra benefits associated to an increase in capacity or improvement of design quality, is complex to valuate. This research will therefore only identify its presence and its potential, but will not try to quantify its value. The value of cost savings on labor is a more reasonable and straightforward calculation. The total value of cost savings is a comparison between business-as-usual and business with software automation. It becomes a trade-off between the cost of software development and maintenance vs. cost of labor. Some critical premises underlying the calculations are:

- one proposal takes 12 Man Months (MM) on average
- six proposals for box structures per year[6]
- engineering labor cost 75€/hr. or 10k€/MM

The calculation of software development cost was based on actual measurements during the various phases of the development cycle[7], actual license costs and an estimate for maintenance. Extra hardware costs are assumed to be absent (existing hardware is adequate). Important premises are:

- software is developed in-house
- programmer labor cost 75€/hr. or 10k€/MM
- initial license costs at 60k€, with an 15k€ annual renewal fee (including support)
- continuous software maintenance and improvements 25% of initial development cost

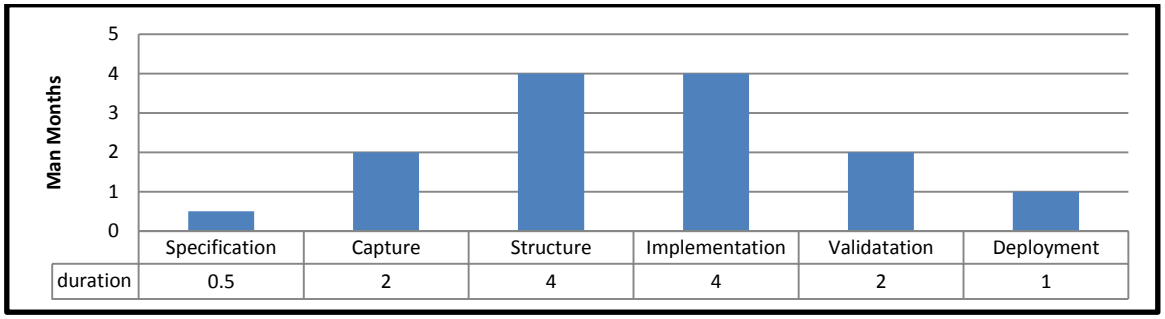

Figure 16: software development cost

The final software framework is conservatively estimated to cut 50% on labor cost. As the annual labor cost for proposals equals 720k€ for business-as-usual, yearly savings of 360k€ can be realized. Figure 16 forms the basis for the non-recurring software development cost of 135k€ (195k€ incl. 1st year licenses). Recurring cost for updates and maintenance is estimated at 48.75k€ per year. Figure 17 provides the expenditure profiles of traditional and KBE-enabled design processes. Assuming the software framework can be completed in one year (spanning six projects), the initial expenditure is higher because of the development/licensing cost. After the first year, initial savings are realized and

[6] Flaps are excluded from this estimate, because they are relatively more complex when compared to rudders, elevators and ailerons for which the current software framework is assumed to be adequate.

[7] All time spent on project definition and scoping, meetings, interviews and programming effort (by three of the authors) have been accurately measured throughout the process.

the break-even point is reached at termination of the 10th project or 20 months from the start of the programme. A second scenario is shown where the software release process is phased. By targeting different releases on focused aspects of the overall design process and with clever scheduling, earlier savings can be realized. Assuming a linear savings profile, first gains are expected during the 7th project or after 14 months. Total savings (phased) after 2, 3, 4 or 5 years are estimated at 0.51M€ [ROI = 109%], 0.87M€ [ROI = 197%], 1.29M€ [ROI = 269%] and 1.59M€ [ROI = 308%], respectively. The high Return on Investment (ROI) justifies the initial risk of investing.

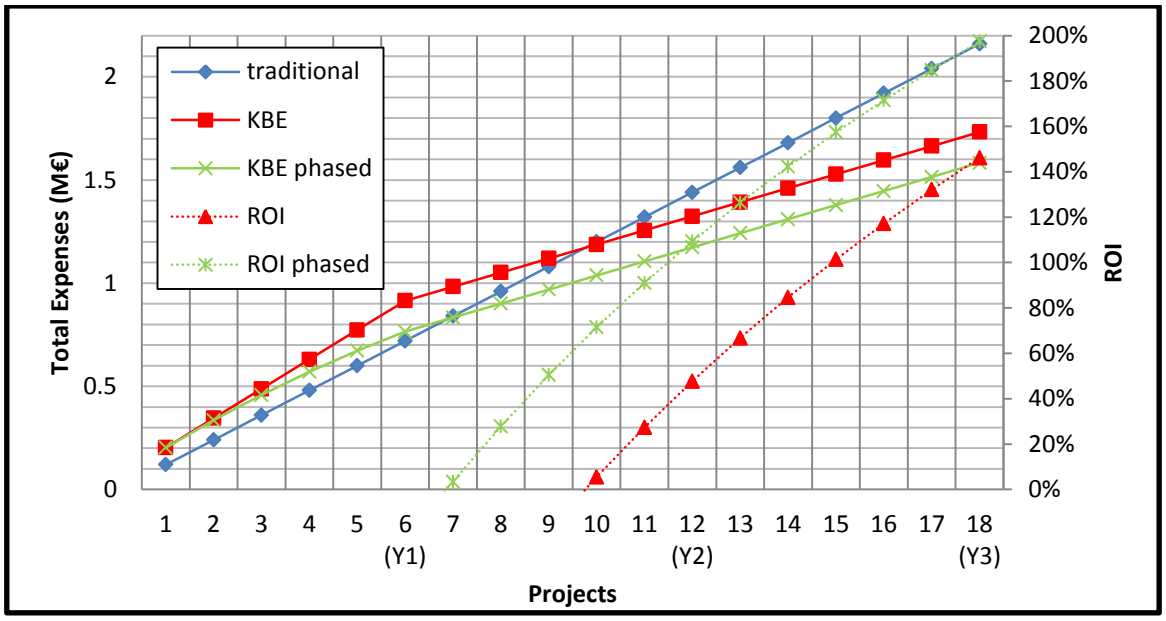

Figure 17: comparison of total expenses between traditional and KBE-enabled design

9.2 Programming Effort

This section will break down the total programming effort for the overall software framework on a per-module basis. The specialization / categorization of which may help in providing a concrete, more extensive insight into the most time-consuming aspects of developing an MDO framework and the requirements that this poses for (next-generation) MDO software development platforms. Inspired by Halstead's complexity measure [12], programming Effort is defined here as the product of Difficulty and Volume,

$$E = D \cdot V \tag{9.1}$$

In contrast to Halstead, both concepts are not rationalized on the basis of measurable software properties (operations and operands), because the pure software focus is too narrow and does not cover the knowledge engineering activities involved in KBE application development. With a lack of alternative quantification schemes, Difficulty in this research is a subjective measure of the complexity experienced during the complete software development process. It starts with problem description, goes through formalization and ends with programming. Complexity during this research has manifested itself in the number of iterations involved in a total programming effort. Many iterations indicate that the formalization of the problem wasn't straightforward. Typically, a procedural script that reads something from or writes something to another data source is easily understood, implemented and hence, perceived easy. On the other hand, an generic object-oriented decomposition of a product model, is observed as a difficult task. The Volume measure is directly related to lines of code. Logically, something that is both difficult and requires a lot of code, requires the biggest Effort.

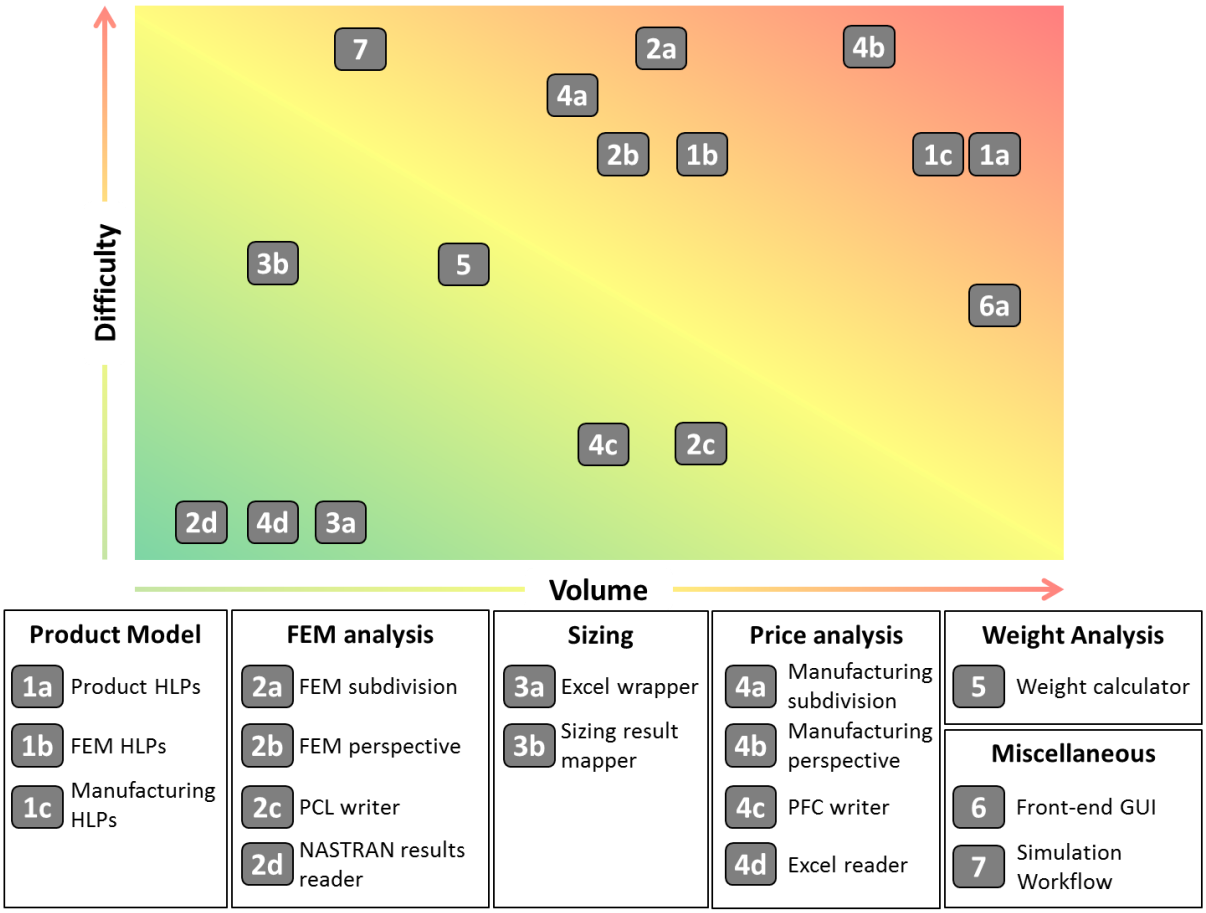

Figure 18: software complexity expressed on the basis of volume and difficulty

Figure 18 shows 16 elementary developments steps that were taken in this research. They are logically grouped conform their discipline. The figure shows that especially the HLP definition process and discipline-specific model generation required the highest efforts. The set of HLP primitives (1a, 1b, 1c) has to result in a generic enough model, typically a highly iterative process. A KBE system language should make it easy to model individual objects and associations between them. A language that is high-level, object-oriented and that hides explicit typing can increase effectiveness. Both subdivision steps involved numerous rule-based geometry manipulations (2a, 4a), where some debugging steps where necessary to robustly handle geometry variations. Hence, a KBE system should build upon a powerful geometry kernel. Moreover, a KBE system should have high computational performance. Aspects like dynamic dependency tracking, lazy evaluation, runtime value caching and compilation are imperative in light of computation time. These observations usually render CAD-centric or MATLAB-like approaches useless. The generation of the discipline-specific perspectives (2b, 4b) involved a lot of engineering rules (especially the manufacturing model). In this respect, it is important that next generation KBE platforms support a model-driven software engineering approach, in which engineering knowledge can be discovered and stored outside the software implementation. This will capture knowledge independent of a KBE platform, increases transparency and traceability. Automatic code generation might lower the perceived programming threshold. The development of a Graphical User Interface (GUI) for this research has turned out to be a voluminous activity, not per se a very difficult one (6a). KBE systems should provide high-level language driven functionality to easily set-up a user interface and define how the user interacts with the product model. In this research, GenDL's General-purpose Web Language (GWL) package has been used to create web interfaces for the MMG. The set-up of a simulation workflow (7) is difficult since it needs a precise definition of the optimization problem. Once the definition is settled, Optimus makes the software workflow development highly interactive, user-friendly and therefore a straightforward

exercise. It must be noted that the already custom-developed Optimus-GenDL interface significantly lowered development time. Finally, on the other side of the extreme are data translators and interfaces with foreign data files or heterogeneous software tools (2d, 3a, 4d). Mostly because these scripts are rather procedural in nature and not dependent on formal knowledge capturing, few iterations were required. However, it was only straightforward because the GenDL platform is built upon a general-purpose programming language. For such languages a vast set of 3rd party packages (often open-source) are available, like the RDNZL package that was used for Excel interfacing. Only a general-purpose programming language based KBE platform can leverage these existing software developments.

10 Conclusions and Future Work

On the basis of Knowledge-Based Engineering and Simulation Workflow Management is has been shown that it's possible to develop a software framework suitable for Multidisciplinary Design Optimization of aircraft box structures. A KBE approach has been chosen over the traditional CAD-oriented approach due to the complexity of the product in terms of parameterization and the range of configurations under consideration. Both product and process knowledge has been captured inside the Aircraft Box Multi-Model Generator to automatically generate the master geometry of a box structure and derive discipline-specific models from there. SFWM automates the software workflow and turns the execution of the KBE application and FEM, sizing, weight and price analysis modules into one seamless process. The next step is to run the actual optimization process. To this end, an algorithm will have to be selected that can deal with a high amount of discrete variables and dynamic updating of the optimization problem definition. When the first MDO results are produced, comparisons will be made against actual G650 rudder design data. Finally, one or more updates will be implemented until the original goals set out in the use case, are achieved.

References

[1] La Rocca, G., *Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design*, Advanced Engineering Informatics, vol. 26, no. 2, pp. 159-179, Apr. 2012.

[2] van Dijk, R.E.C., d'Ippolito, R., Tosi, G., La Rocca, *G. Multidisciplinary Design and Optimization of a Plastic Injection Mold using an Integrated Design and Engineering Environment*, 2011. Available at: http://repository.tudelft.nl/view/ir/uuid:26d5c229-c770-4080-82af-b12ae97a2a0b/

[3] Genworks International, *Genworks GenDL* Manual. Available at: http://www.genworks.com

[4] NOESIS Solutions N.V., *Optimus*. Available at: http://www.noesissolutions.com/Noesis/node/19

[5] J. Roskam, *Airplane Design*. Published by the author as an 8 volume set, 1985-1990.

[6] R. W. Hess, H. P. Romanoff, "Aircraft Airframe Cost Estimating Relationships," Rand Corp. Report R-3255-AF, 1987. Available at: http://www.rand.org/pubs/notes/2005/N2283.1.pdf

[7] D. P. Raymer, "Aircraft Design: A Conceptual Approach, 3rd Edition," AIAA Education Series, 1999.[8]

[8] S. M. Haffner, *Cost Modeling and Design for Manufacturing Guidelines for Advanced Composite Fabrication*, PhD. Thesis, Massachusetts Institute of Technology, 2002. Available at: http://dspace.mit.edu/handle/1721.1/8138?show=full

[9] A. H. van der Laan, *Knowledge Based Engineering Support for Aircraft Component Design*, PhD. Thesis, 2007.

[10] H. Wang, X. Zhao, G. La Rocca, R. Curran and M.J.L. van Tooren, *Parametric modeling of fuselage panels for structural analysis and cost estimation*, Proceedings of the 3rd Aircraft Structural Design Conference, Delft, 2012.

[11] R. Curran, S. Raghunathan, M. Price, *Review of aerospace engineering cost modeling The genetic causal approach. Progress in Aerospace Sciences*, vol. 40, pp. 487-534, doi:10.1016/j.paerosci.2004.

[12] Halstead, Maurice H., *Elements of Software Science*. Amsterdam: Elsevier North-Holland, Inc.. ISBN 0-444-00205-7, 1977.

[13] E. Weitz, *RDNZL - A .NET layer for Common Lisp*, 2010. Available at: http://weitz.de/rdnzl/

[14] Wim J.C. Verhagen, Pablo Bermell-Garcia, Reinier E.C. van Dijk, Richard Curran, *A critical review of Knowledge-Based Engineering: An identification of research challenges, Advanced Engineering Informatics*, Volume 26, Issue 1, January 2012, Pages 5-15, ISSN 1474-0346, 10.1016/j.aei.2011.06.004.