# THE DESIGN AND ENGINEERING ENGINE. TOWARDS A MODULAR SYSTEM FOR COLLABORATIVE AIRCRAFT DESIGN

**G. La Rocca, T.H.M. Langen, Y.H.A. Brouwers**
**Delft University of Technology, The Netherlands**
*g.larocca@tudelft.nl*

**Keywords**: *Design and Engineering Engine, aircraft design, collaborative design, knowledge based engineering, multidisciplinary design optimization*

## Abstract

*This paper describes the architecture of the Design and Engineering Engine (DEE), a modular, loosely integrated software system able to support conceptual and multidisciplinary design optimization of both conventional and novel aircraft configurations. In particular, it elaborates on the functionalities and the current state of development of two key components, namely the Initiator and the Multi Model Generator. The first module is responsible for the aircraft conceptual design process, but makes use of Knowledge Based Engineering and optimization techniques to overcome the limits of classic textbook methods, particularly when dealing with novel configurations. The Multi Model Generator (MMG) is a Knowledge Based Engineering application that, starting from the set of design parameter values generated by the Initiator, can automatically generate the geometry model of many different aircraft configurations, plus the relative abstractions required to use a distributed and heterogeneous set of analysis tools, commercial and proprietary, high and low fidelity. The modular and open architecture of the DEE provides scalability and adaptability to different design problems. In particular, the MMG can be used to feed analysis tools provided by external parties and located on geographically non collocated machines, thereby enabling truly distributed and collaborative design.*

## 1. Introduction

The Advisory Council for Aeronautical Research in Europe, similarly to NASA in the United State, has devised a challenging roadmap to the help the aerospace industry stepping into a new age of sustainable growth [1-3]. Yet, it seems impossible to achieve the set objectives without major improvements in the way aircraft are designed today. New tools and methods are required to ease distributed and collaborative design, increase productivity of the scarce intellectual resources and better support the decision making process. Such tools should be able to improve the performance of current designs, as well as to support the development of novel aircraft configurations.

In the last decades, several new and "unorthodox" aircraft configurations, like the blended wing body and joint wings aircraft, have been proposed by visionary designers [4, 5]. However, conventional design methods appear inadequate because of the very strong and not always evident disciplines coupling featured by these *highly integrated vehicles* [6], and because of the lack of reference and statistical data. Multidisciplinary design optimization (MDO) is claimed to be the way forward, both to improve current configurations, and to support clean sheet designs [6-8]. Still, the development of design systems able to effectively support the MDO approach is an open challenge.

There are continuous attempts by industry and academia to develop complex integrated design tools to cover the whole aircraft design cycle, from drafting to high fidelity multidisciplinary analysis and optimization [9]. Eventually, these systems turn useful to address only one part of the design process, e.g., the conceptual design phase, where only low fidelity analysis tools or simple semi-empirical methods are generally employed. Besides, they are difficult to scale up and maintain and, above all, they are unsuitable for collaborative design initiatives. At the same time, discipline experts want to use (and keep on developing) their own trusted analysis tools, which, eventually, need to be integrated or federated in larger multidisciplinary design frameworks [10, 11].

To address the aforementioned challenges, an advanced, modular design framework is being developed at the Technical University of Delft, called the Design and Engineering Engine (DEE). The overall architecture of the DEE is described in Section 2, where Subsections 2.1-2.3 provide details on some of its key modules, namely the Initiator and the Multi Model Generator. Section 3 addresses some specific issues related to data exchange for distributed collaborative design. In section 4, the value of the proposed approach and its effect on the aircraft design process are briefly discussed. Conclusions are provided in section 5.

## 2. The Design and Engineering Engine architecture

The DEE is an advanced design system concept to support and accelerate the design process of aircraft and/or aircraft sub-systems, through the automation of non-creative and repetitive design activities [7, 12]. It consists of a multi-disciplinary collection of design and analysis tools, able to interface and exchange data and information. See Fig. 1.

The main components of the DEE are the followings:

- The **Multi Model Generator** (MMG), which is a Knowledge Based Engineering (KBE) application, developed with the twofold intent of providing designers with aircraft generative modeling capability, and supporting multidisciplinary analysis by extensive automation of the model preprocessing activities. Details in Subsection 2.3.

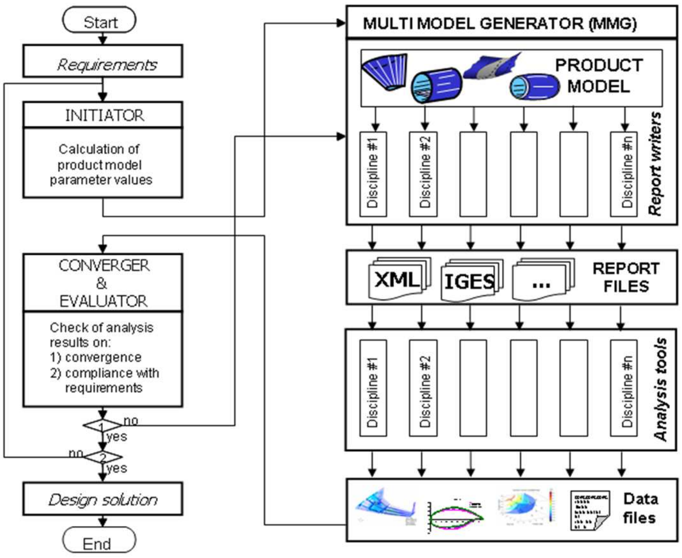


Fig.1. Paradigm of the Design and Engineering Engine

- The **Initiator**, which consists of a set of more sizing modules that, starting from a limited set of top level requirements, can provide the MMG with an initial set of parameter values to start the generative modeling process. In fact, the MMG can instantiate an aircraft model only based on a given set of input parameters values, but it does not have any knowledge to select/calculate those values autonomously. More details in Section 2.1.
- A **suite of analysis tools**, which can be low and high fidelity analysis tools (e.g., panel codes and CFD), either in-house developed or off the shelf (e.g., VS-Aero and NASTRAN). The set of analysis tools is not fixed a priory and can vary according to the design case at hand.
- The **Converger&Evaluator** module (generally an off-the-shelf optimizer), whose tasks include checking the convergence of the various analysis tools (e.g. the flow solver), evaluating whether the performance/characteristics of the design meet the set objectives, and defining the next parameter set when running an optimization process.
- The **communication framework**, represented in Fig. 1 by the set of connectors linking the various DEE components, which takes care of the data and information flow

between the various design and analysis tools and enables the overall design process sequence.

In order to join the DEE, any software component must be able to operate autonomously, possibly in *batch mode* (although interaction is required for the initialization phase of the design) and expose an adequate input/output interface. Data can be exchanged directly between tools, or, as currently investigated, recorded and distributed via the centralized CPACS data structure, developed by DLR [13] (see Section 3).

## 2.1. The DEE initiator

The current DEE Initiator consists of a main MATLAB application [14], supported by a KBE application for preliminary fuselage sizing and configuration [15]. In fact, this KBE application is a component of the MMG, which can be operated independently (details in Subsection 2.2).

The main MATLAB application is able to generate a baseline aircraft design, starting from a limited set of top level requirements, such as payload size and arrangement, range, cruise speed, takeoff and landing field length. Apart from conventional turboprop and turbofan aircraft, the Initiator can deal with joint-wing (or box-wing) configurations [5]. Extensions are currently under development to address also the conceptual design of three lifting surfaces and blended wing body aircraft [16].

The Initiator implements some of the classical aircraft synthesis methods available in literature [17, 18]. Furthermore, it makes use of simple geometry models generated on the fly, a vortex lattice aerodynamic simulation tool and an optimization toolbox. These "extra ingredients" are supposed to make designers much less dependent on statistics and crude approximations, and help them to quickly iterate towards an optimum baseline design. To this purpose, wetted surfaces and volumes are directly extracted from the geometrical models and the aerodynamic derivatives are computed using simulations. Finally, optimization techniques are used to fine tune and improve the given aircraft configuration, while guaranteeing fulfillment of customer and airworthiness requirements. The use of optimization techniques is particularly helpful to deal with the design of joined-wing configurations, where the number of design parameters is larger than for a conventional aircraft and the aerodynamic reciprocal influence of the front and rear wing is difficult/impossible to assess using classical semi-empirical methods.

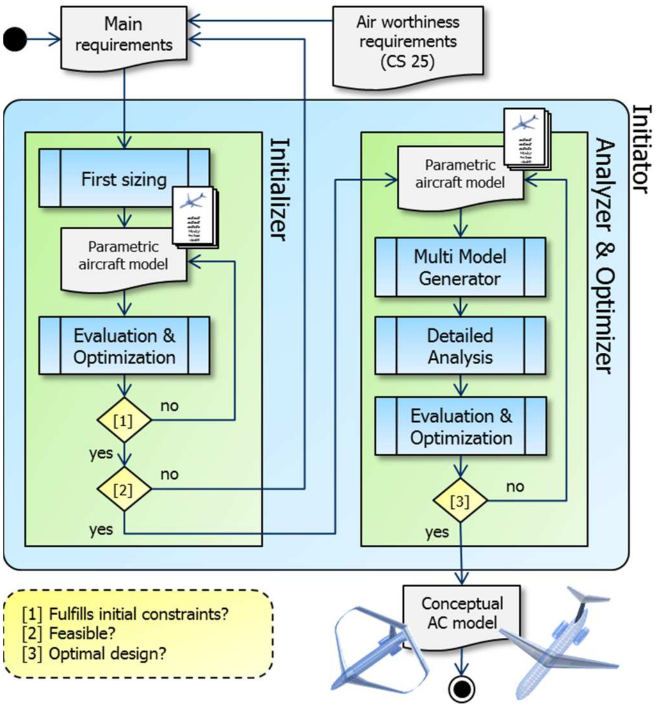


Fig. 2. The DEE Initiator structure

The similarity of the Initiator and DEE architecture is noteworthy. As a matter of fact, the Initiator is a kind of DEE itself. As shown in figure 2, the Initiator contains an initialization module, a geometry model generator, some analysis modules and an optimizer. The "Initiator's initiator", called *Initializer*, has the task of deriving a first aircraft guesstimate, based on pure statistical data. To this purpose a large and extensible aircraft data base has been developed, which is automatically accessed by the Initiator. Before proceeding with any further analysis, wing loading and thrust weight ratio are automatically adjusted using an optimization routine, to make sure the aircraft design point satisfies typical top level requirements, such as takeoff and landing field length, climb rate and gradients at OEI conditions, etc.

The Initiator geometry modeler makes use of the MATLAB (limited) geometry modeling

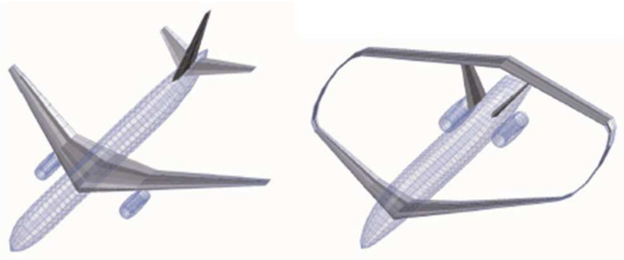


Fig. 3. Geometry models generated by the Initiator for a conventional and a joined-wing aircraft

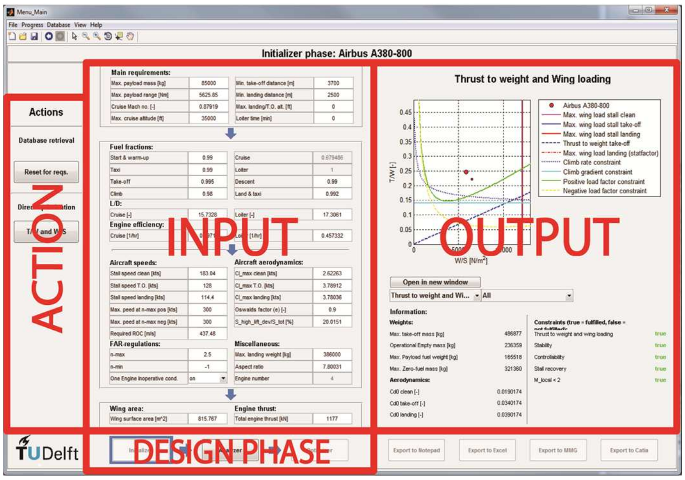
and manipulation capabilities to create simple aircraft models (see two examples in fig. 3), where volumes, areas, distances, etc., are extracted to feed the implemented semi-empirical analysis and sizing methods.

These geometry models are used also to feed TORNADO, an open source vortex lattice method (VLM) suitable for conceptual design purpose. Since TORNADO is natively a MATLAB application, it was straightforward to embed it in the Initiator. Although TORNADO is a low fidelity analysis tool, it allows the Initiator to account on more physics based aerodynamic results than those otherwise assumed based on statistics and generally only valid for conventional aircraft configurations. Some other of the Initiator analysis modules include a class I and class II weight estimation tool, a module for parasite drag estimation and a module for stability & control.

A genetic algorithm optimizer has been developed on purpose to endow the Initiator with robust optimization capabilities. The Optimizer allows the designers to assess the impact of various objectives and constraints on the final design of the aircraft and its performances. Besides, the optimizer (in addition to the VLM module) is particularly useful for the initial sizing of joined-wing systems, where the relative positioning of the front and rear wing and their relative lift distributions need to be properly set to achieve proper stall behaviour and exploit the Prandtl's *best wing system concept* for minimum induced drag [5].

An advanced GUI (see one screenshot in Figure 4) allows the designer to access all the functionalities of the Initiator, edit default values, overwrite calculation results (when more reliable values are available from other sources) and set up different multi objective optimization problems. Functionalities are in place to export all the generated values (geometry, weights, performance parameters, etc.) in form of Excel tables, or XML files, as further elaborated in Section 3. All the generated plots (payload-range diagrams, wing loading/thrust loading diagrams, etc.) can also be exported for reporting use.

To guarantee scalability and maintainability, the Initiator has been designed with a strongly modular architecture. The various computational modules never exchange data directly with each other, but communicate only through one common data layer. This makes the data flow very transparent, and enable a plug and play approach for new or improved modules.



Initializer phase: Airbus A380-800

**Main requirements:**

| | | | |
|---|---|---|---|
| Max. payload mass [kg] | 85000 | Min. take-off distance [m] | 3700 |
| Max. payload range [Nm] | 5625.85 | Min. landing distance [m] | 2500 |
| Cruise Mach no. [-] | 0.87919 | Max. landing/T.O. alt. [ft] | 0 |
| Max. cruise altitude [ft] | 35000 | Loiter time [min] | 0 |

**Fuel fractions:**

| | | | |
|---|---|---|---|
| Start & warm-up | 0.99 | Cruise | 0.679486 |
| Taxi | 0.99 | Loiter | 1 |
| Take-off | 0.995 | Descent | 0.99 |
| Climb | 0.98 | Land & taxi | 0.992 |

**L/D:**

| | | | |
|---|---|---|---|
| Cruise [-] | 15.7328 | Loiter [-] | 17.3061 |

**Engine efficiency:**

| | | | |
|---|---|---|---|
| Cruise [1/hr] | [illegible] | Lo [1/hr] | 0.457332 |

**Aircraft speeds:**

| | |
|---|---|
| Stall speed clean [kts] | 183.04 |
| Stall speed T.O. [kts] | 128 |
| Stall speed landing [kts] | 114.4 |
| Max. peed at n-max pos [kts] | 300 |
| Max. peed at n-max neg [kts] | 300 |
| Required ROC [m/s] | 437.48 |

**Aircraft aerodynamics:**

| | |
|---|---|
| Cl_max clean [kts] | 2.62263 |
| Cl_max T.O. [kts] | 3.78912 |
| Cl_max landing [kts] | 3.78036 |
| Oswalds factor (e) [-] | 0.9 |
| S_high_lift_dev/S_tot [%] | 20.0151 |

**FAR-regulations:**

| | |
|---|---|
| n-max | 2.5 |
| n-min | -1 |
| One Engine Inoperative cond. | on |

**Miscellaneous:**

| | |
|---|---|
| Max. landing weight [kg] | 386000 |
| Aspect ratio | 7.80031 |
| Engine number | 4 |

**Wing area:**

| | |
|---|---|
| Wing surface area [m^2] | 815.767 |

**Engine thrust:**

| | |
|---|---|
| Total engine thrust [kN] | 1177 |

**Information:**

**Weights:**

| | |
|---|---|
| Max. take-off mass [kg] | 486877 |
| Operational Empty mass [kg] | 236359 |
| Max. Payload fuel weight [kg] | 165518 |
| Max. Zero-fuel mass [kg] | 321360 |

**Aerodynamics:**

| | |
|---|---|
| Cd0 clean [-] | 0.0190174 |
| Cd0 take-off [-] | 0.0340174 |
| Cd0 landing [-] | 0.0390174 |

**Constraints (true = fulfilled, false = not fulfilled):**

| | |
|---|---|
| Thrust to weight and wing loading | true |
| Stability | true |
| Controllability | true |
| Stall recovery | true |
| M_local < 2 | true |

Fig. 4. Screenshot of the Initiator GUI. Details of the Initializer interface and its main panes to edit input values, check the results of the Class I weight estimation and the preliminary wing loading and thrust matching.

## 2.2. The fuselage configurator

The fuselage model generated by the abovementioned MATLAB-based Initiator (actually by the Initializer) does not include any interior detail and consists of a simple surface model, whose main dimensions are purely based on statistics. The length and the shape of the fuselage cross sections are just derived extrapolating/interpolating values of existing aircraft with similar mission requirements.

While this approach is very fast and can produce decent results for conventional aircraft, it does not allow the designer to make decision on the fuselage design and judge the effect of different payload accommodation. Also, it does not allow a proper estimation of the aircraft center of gravity during loading/unloading of passengers and freight, which is crucial information to address the stability and controllability of the aircraft, as well as for the estimation of trim drag.

In order to produce reliable and more detailed fuselage designs, including the main interior items, a dedicated KBE application has been developed using the commercial platform GDL [19]. In fact, this KBE fuselage configurator is one module of the DEE Multi Model Generator, whose general capabilities and architecture are described later in Subsection 2.3. The user of the Initiator can decide whether to use the simplified statistic based fuselage sizing approach or the MMG fuselage configuration module. In the latter case, the specific geometrical models of the fuselage aerodynamic surface and interiors are automatically generated, on the basis of a limited amount of top level requirements (e.g., number of passengers and classes, number and type of unit load devices) and making use of the so-called *inside-out* design approach. Rules are used first to define the most convenient distribution of passengers, freight, pilots and flight attendants and then to envelope them all inside an aerodynamically reasonable shape.

The CST curve parameterization method proposed by Kulfan [20] and an optimization routine are used to fit the best cross sections (circular, elliptical, double bubble or quasi-free form) around the payload, which, in turn, is distributed such to obtain favorable values of fuselage slenderness (i.e., the ratio of max fuselage length and cross section diameter).

Designers' preferences, sets of editable data concerning interiors items, and various rules to guarantee compliance to airworthiness regulations (e.g., number, type and position of emergency exits, minimum size for aisle(s) and clearances, etc.) are used by the KBE application to derive the final design. One example for a wide body aircraft is shown in figure 5.

In order to validate the tool, the payload requirements of several existing aircraft have been used as input for the KBE fuselage configurator. As shown in table 1, the obtained designs nicely match the size of existing aircraft, both wide body and single aisle.

Next to the geometrical definition of the

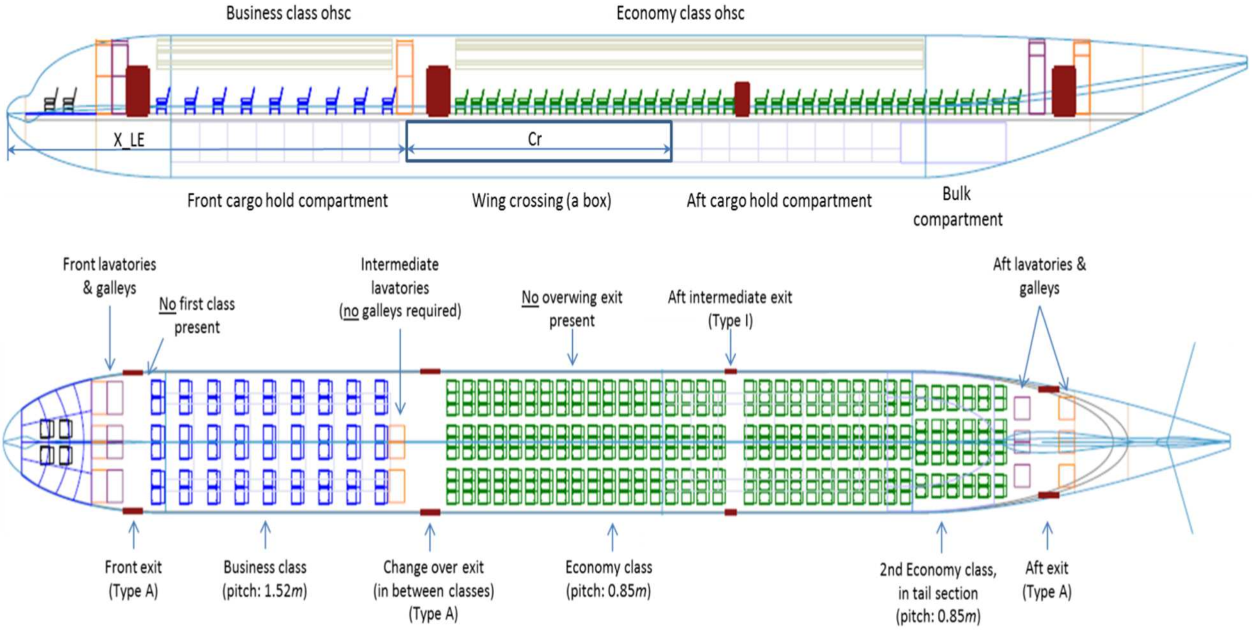


Fig. 5. Example of fuselage layout generated by the KBE fuselage configurator.

| Airplane | Width [m] | Height [m] | Length [m] |
|---|---|---|---|
| A300-600 | 5.62 vs 5.64 (- 0.4%) | 5.62 vs 5.64 (-0.4%) | 51.54 vs 54.08 (-4.7%) |
| A320-100 | 3.80 vs 3.95 (-3.8%) | 4.05 vs 4.14 (-2.2%) | 38.64 vs 37.57 (+2.8%) |
| A340-600 | 5.71 vs 5.64 (+1.2%) | 5.71 vs 5.66 (+0.9%) | 73.10 vs 73.46 (-0.5%) |
| A350-900 | 5.88 vs 5.96 (-1.3%) | 5.77 vs 6.09 (-5.3%) | 67.09 vs 66.89 (+0.3%) |
| | | | |
| ATR 42-500 | 2.75 vs 2.85 (-3.5%) | 2.75 vs 2.85 (-3.5%) | 21.94 vs 22.67 (-3.2%) |
| ERJ 145 | 2.32 vs 2.28 (+1.8%) | 2.32 vs 2.28 (+1.8%) | 27.70 vs 27.93 (-0.3%) |
| MRJ 70 | 2.83 vs 2.96 (-4.4%) | 2.83 vs 2.96 (-4.4%) | 33.40 vs 33.60 (-0.6%) |
| Saab 340 | 2.30 vs 2.31 (-0.4%) | 2.30 vs 2.31 (-0.4%) | 21.04 vs 19.70 (+6.8%) |

Table 1. Verification of the fuselage initiator: generated model size vs. actual aircraft size (based on same payload requirements)

fuselage, the KBE configurator generates the loading diagram of the aircraft. The center of gravity range for all the possible loading conditions is computed and fed back to the MATLAB-based Initiator, where it is used for stability and control calculations.

The set of fuselage cross sections (supported by belly, crown and side longitudinal curves) generated by the KBE fuselage configurator, actually, represents the input required by the MMG to instantiate the fuselage High Level Primitive, as it will be clarified in Subsection 2.3. Fig. 6 shows an example of aircraft model eventually generated by the MMG, inclusive of all the interior details.

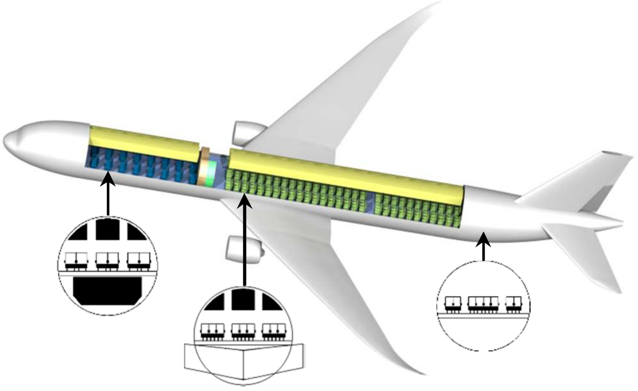


Fig. 6: example of passenger aircraft design generated by the Multi Model Generator, using Initiator results (based on A350-900 XWB requirements).

## 2.3. The DEE Multi Model Generator

The Multi-Model Generator (MMG) is an extensive and extensible KBE application developed with a twofold intent:

1. To provide designers with a parametric modeling environment to define generative models of conventional and novel aircraft configurations
2. To feed various analysis tools with dedicated aircraft model abstractions, as required for the verification of the generated design.

To meet these objectives, two types of functional blocks have been developed, which constitute the main ingredients of the MMG: the High Level Primitives (HLPs) and the Capability Modules (CMs). They take care of the intent stated in bullet point 1 and 2 respectively.

The set of High Level Primitives defined so far, includes the Wing-part, the Fuselage-part and the Engine primitive. These three primitives can be figured out as a suite of advanced LEGO blocks that designers can manipulate and assemble to build up an extremely large number of aircraft configurations and variants, including novel air vehicle concepts. This modeling concept is schematically illustrated in Fig. 7, where it is shown how the same Wing-part primitive can be re-used more times to model the wing, the winglets and the empennages of a conventional aircraft, as well as the wing and the center body section of a blended wing body.

In practice, the HLPs are *classes* defined using the object oriented programming language provided by the employed KBE system [21]. Each class can be instantiated many times by providing different attributes values. Designers can control the value of these attribute, as well as the number and type of HLPs to be instantiated, by means of the MMG input file. In this way, various aircraft configurations can be automatically generated and then

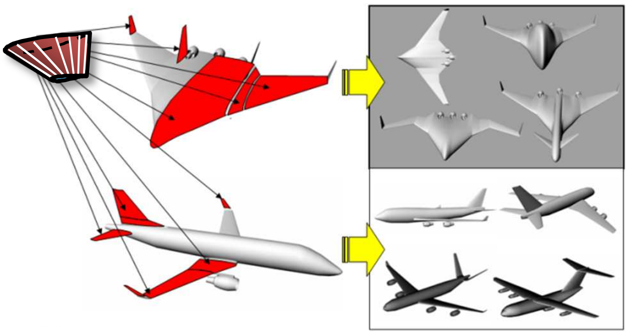


Fig. 7: The HLP approach to model different aircraft configurations and their parametric variants.

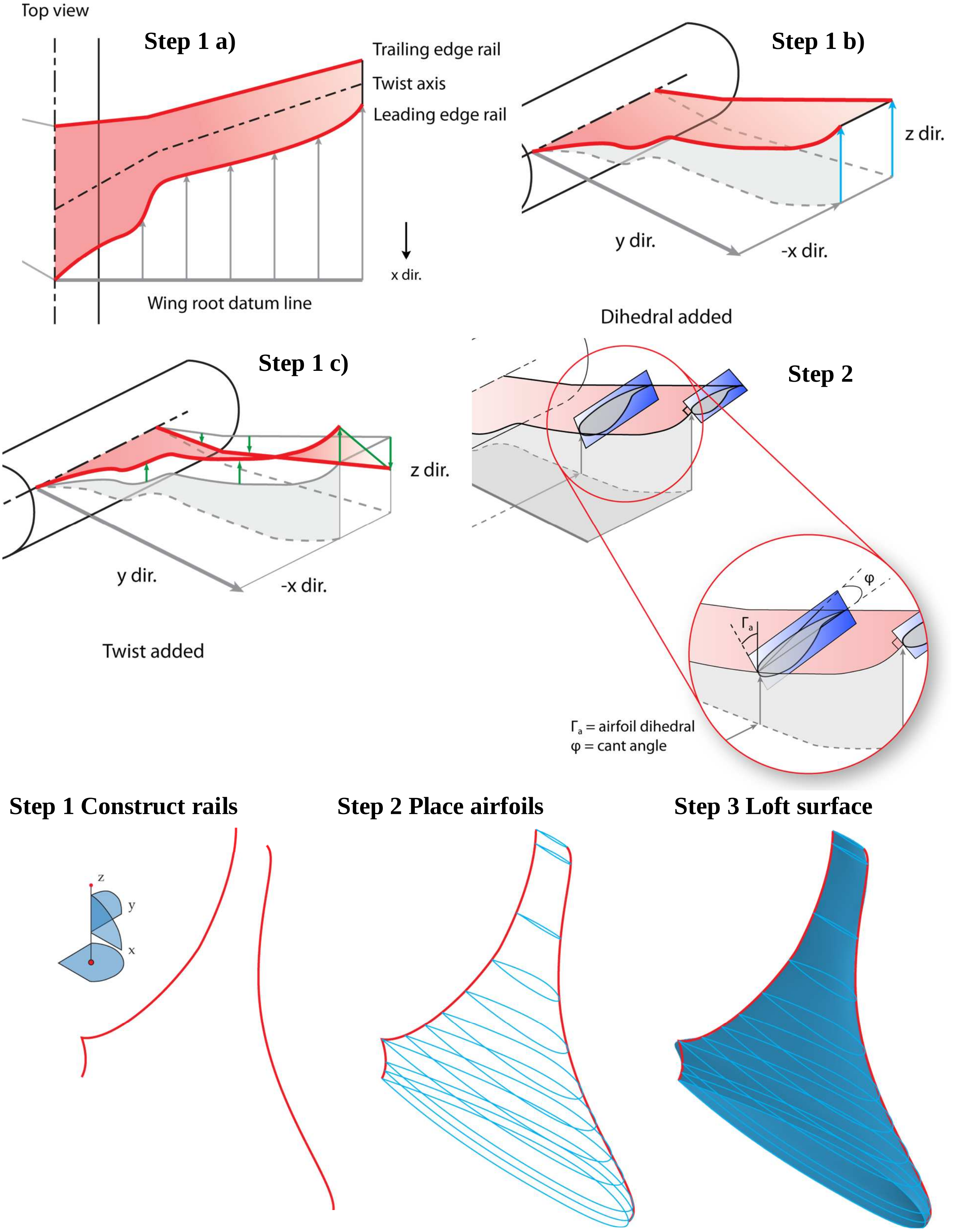


Fig. 8. Definition of the Wing-Part HLP. Two curvilinear and not necessary continuous rails are defined and used to “hinge” airfoils at any angle. Twist, dihedral and sweep angle distributions are not necessarily linear.

stretched/morphed into an infinite amount of design variants.

The level of fidelity, flexibility, accuracy and detail of the geometry models generated by the MMG is far larger than the simple ones generated by the MATLAB-based Initiator. For example, they include the internal structural layout and a representation of the main systems. In case of the fuselage, the interiors are also included, as described in Subsection 2.2. In fact, the MMG is supposed to satisfy the needs of the *preliminary* design phase, hence it must support the use of high fidelity analysis too, such as CFD and FE codes.

A first version of the MMG, based on the ICAD KBE system (now out of the market), was presented in previous publications [7, 22-24]. A new generation MMG is currently under development using the GDL system. Although this new application is based on the same modeling principles (i.e., High Level Primitives and Capability Modules) of the obsolete ICAD MMG, it incorporates new features, such as high lift devices [25] for instance, and makes use of more advanced modeling techniques to enhance modeling flexibility and ease of use. The fuselage interiors configuration capability described in Subsection 2.2, is also one of the new MMG capability. The modeling approach for lifting surface has been reviewed with respect to the ICAD version of the MMG. A few details of the latest Wing-part HLP modeling approach are shown in fig. 8 [26].

Once the geometry model of the given aircraft has been instantiated, the Capability Modules (CMs) take care of generating the various discipline abstractions for the DEE analysis tools. Different CMs have been developed to take care, for example, of the model preprocessing for FEM analysis, or for CFD analysis. To automate these preprocessing activities, acknowledged to be time expensive and repetitive, "model preprocessing knowledge" has been acquired by discipline experts and then encoded in the Capability Modules, making use of the KBE programming language. In this way, CMs can systematically apply the experts' best practices and automate the generation of models for a broad range of low and high fidelity analysis tools, both proprietary and commercial off the shelf.

Fig. 9 shows examples of possible high lift devices configurations that can be modeled with the MMG and their relative lift curves, as obtained using CFD simulation (MSES) and semi empirical methods (ESDU) [25].

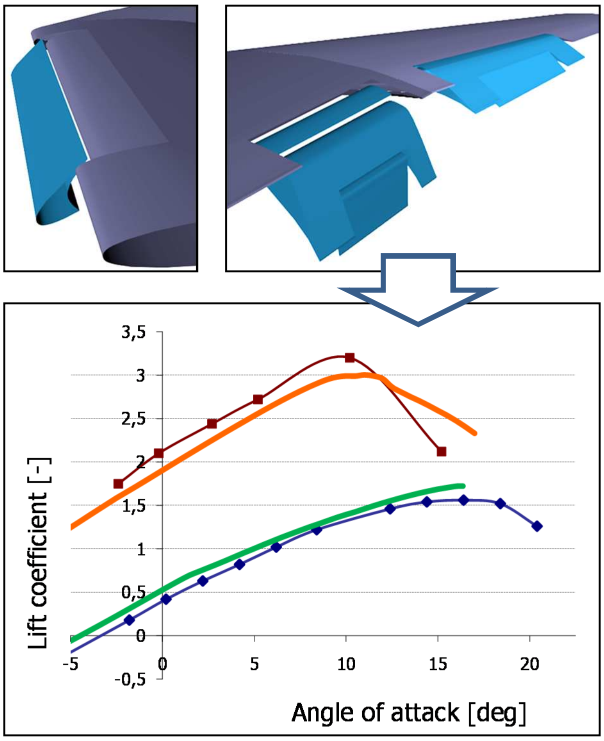


Fig. 9. Examples of MMG generated multi-element wings (slat and triple slotted flaps). Different Capability Modules take care of deriving from the same master geometry model, the abstractions needed for CFD simulations (MSES) and semi empirical methods (ESDU).

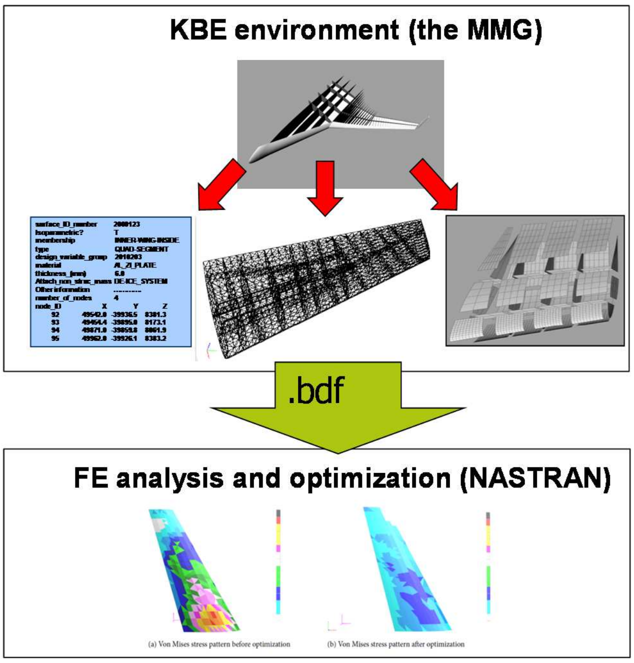


Fig. 10. The KBE system surface tessellation capability is used to generate suitable grids for FE structural analysis directly within the MMG. A Capability Module generates the NASTRAN bulk data deck (.bdf) file..

Fig. 10 illustrates an implemented method to automate the generation of finite elements model for lifting surfaces [26, 27]. A recently implemented method to automate FEM analysis of fuselage structures is described in [28].

The general use mode of the MMG and its input/output architecture are illustrated in Fig. 11. When the MMG operates within the DEE framework, its input files and fuselage curves repository are produced by the Initiator module described in Subsection 2.2.

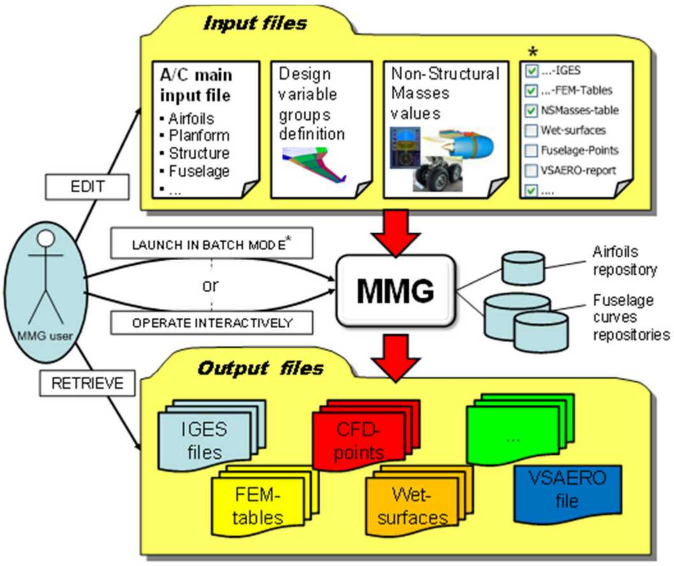


Fig. 11. Operation of the MMG and its input/output architecture.

## 3. A common language to support collaborative design

The capability to perform distributed multidisciplinary design optimization of aircraft, using the set of tools described in the previous sections (or their predecessors, or other parties counterparts), has been partly demonstrated in previous projects [7, 10, 22, 23, 29]. However, the level of flexibility of the assembled design systems was generally low. Especially in case of complex design problems, the management of data transformation and exchange was often based on the use of ad-hoc solutions, not always transparent and difficult to re-use beyond the design case at hand. As a consequence, a truly plug-and-play approach was rarely possible and the level of reconfiguration agility was poor. In particular, *welcoming* a third party tool within an existing design framework, came with the usual overhead of defining, building and testing new ad-hoc interfaces and data exchange formats. Although, technically not challenging, practically, this is a critical obstacle for collaborative design.

To this purpose, a collaboration initiative [30] has recently started, involving a number of international partners (TU Delft, DLR, KTH and Stanford University) to assess the advantages of using *a common language for aircraft representation*, namely the DLR developed CPACS (Common Parametric Aircraft Configuration Schema) [13]. In practice, CPACS is an extensive XML schema, which aims at standardizing the way to describe an aircraft (and its operative environment), including, among others, geometry and performance data. The CPACS schema ships with a number of utilities, including data validation tools and visualization plug-ins to generate and inspect the geometry of the described aircraft. Once partners (actually, their tools) are able to read data stored with CPACS and feed their generated results back to CPACS, any collaborative design effort will be largely facilitated.

Fig. 12 shows the currently investigated approach of using CPACS to exchange data between the MATLAB-based Initiator and the KBE MMG, as well as among different design and analysis tools, such as flight mechanics toolboxes, FEM codes, etc. To this purpose, the

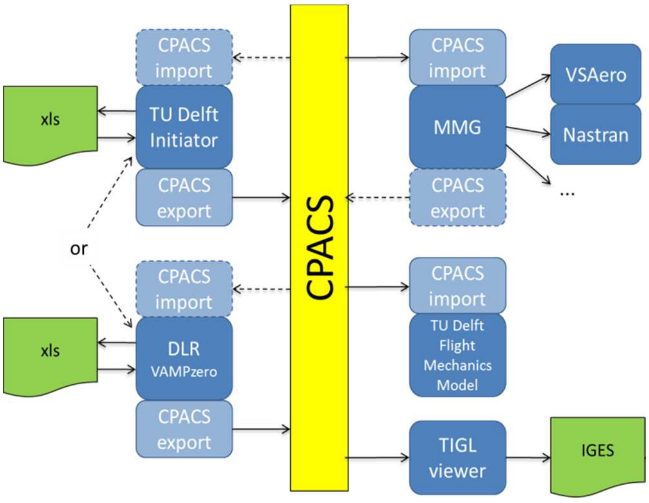


Fig. 12. Different conceptual design (left) and analysis (right) tools collaborating through CPACS

Initiator has been provided with a CPACS export functionality, while the MMG is currently being updated to extract its input data directly from CPACS. Eventually, this strategy should enable, for example, to swap the TU Delft Initiator with the DLR counterpart VAMPzero, Similarly it should facilitate the exchange and sharing of various analysis tools, whose deployment is generally limited by ad-hoc data exchange format.

## 4. Discussion

The presence of the Initiator and the MMG inside the DEE allows blending into one system the design space exploration capabilities offered by the MDO approach with the availability of proven aircraft design and sizing methods. Through the Initiator, a baseline configuration can be first synthesized, using handbook methods and data from reference aircraft, and subsequently fed to the "MDO machine", through the support of the MMG. In this way, the DEE allows a smooth transition *from the conceptual to the preliminary design phase of the aircraft*, where the output of the Initiator can be further detailed by the MMG and improved by means of more accurate analysis tools, without creating gaps in the overall design process. The traditional distinction between conceptual, preliminary and detail design phase appears now rather blurry.

Because of the preprocessing automation provided by the MMG, high fidelity analysis tools can be used earlier in the design process. In general, this has a positive effect on the level of confidence of the designed product. In particular, this is required to lower the development risk of innovative aircraft configurations, for which semi-empirical and statistics based methods are not sufficient and first principle analysis is the only way to go. In this sense, the DEE offers a possible solution to what Lockheed Martin's specialists indicate as "*the need to successfully leverage the best design knowledge available, but push beyond results predestined by heritage databases and empirical correlations* [31]".

## 5. Conclusions and next steps

The concept of Design and Engineering Engine has been discussed in this paper, with particular emphasis to two key modules: the Initiator and the Multi Model Generator.

The Initiator is able to generate very quickly conceptual designs of both conventional and novel aircraft configurations, such as joined-wings. The early and integrated use of optimization techniques, geometry models and aerodynamic simulation tools distinguishes the Initiator from the more conventional aircraft synthesis tools available on the market.

The generative capability of Knowledge Based Engineering has been exploited to implement an advanced aircraft parametric modeling system, the Multi Model Generator, which is able to automate the lengthy and complex preprocessing activities required by high fidelity analysis tools in particular. The MMG includes also a fuselage configurator module, which is based on the inside-out sizing approach used for conventional aircraft fuselages.

Once a baseline design has been generated by the Initiator, it can be remodeled at a higher level of detail and accuracy by the MMG, to enable more sophisticated analysis and support distributed optimization.

Both the Initiator and the MMG are currently being extended and improved in terms of general capabilities, flexibility and robustness. In particular, the Initiator is being extended to support the design of Blended Wing bodies and other non-conventional configurations.

Developments at interface level is also undergoing to make the MMG and the Initiator CPACS compatible, hence easier to deploy in large and distributed computational systems for collaborative aircraft design studies.

## Acknowledgement

The authors like to acknowledge the fundamental contributions of Justin Koning, Tobie van den Berg and Maarten van Hoek to the development of the GDL-based Multi Model Generator. Elements of the Initiator development have been partly sponsored by the CLEANSKY - Joint Technology Initiative (JTI) projects AERODESIGN and WEMACS.

## References

[1] Group of Personalities. European aeronautics: A vision for 2020. in, European Commission, Luxembourg, 2001.

[2] ACARE. Aeronautics and air transport: Beyond vision 2020 (towards 2050) - *background document*. in, www.acare4europe.com, 2010.

[3] ACARE. 2008 addendum to the strategic research agenda. in, www.acare4europe.com, 2008.

[4] Liebeck RH. Design of the blended wing body subsonic transport. *Journal of Aircraft*, 41 (2004).

[5] Frediani A. The Prandtlplane. in: *ICCES04*, Tech Science Press, Madeira, Portugal, 2004, pp. pp 19-31.

[6] Bowcutt K. A perspective on the future of aerospace vehicle design. in: *12th AIAA International Space Planes and Hypersonic Systems and Technologies*, AIAA, Norfolk, Virginia, 2003.

[7] La Rocca G. Knowledge based engineering techniques to support aircraft design and optimization (doctoral dissertation). in: *Faculty of Aerospace Engineering*, TU Delft, Delft, 2011.

[8] Vandenbrande J, H., Grandine T, A. and Hogan T. The search of the perfect body: Shape control for multidisciplinary design optimization. in: *44th AIAA Aerospace Science Meeting and Exhibit*, AIAA, Reno, Nevada, 2006.

[9] Butler R, Lillico M, Hansson E and van Dalen F. Comparison of mdo codes for use in conceptual and preliminary aircraft wing design. in: *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, St. Louis, MO, 1998.

[10] Morris AJ, La Rocca G, Arendsen P, Laban M, Voss R and Hönlinger H. Mob - a european project on multidisciplinary design optimisation. in: *24th ICAS Congress*, ICAS, Yokohama, Japan, 2004.

[11] Bartholomew P. The role of mdo within aerospace design and progress towards an MDO capability through european collaboration. in: *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, St. Louis, MO, 1998.

[12] La Rocca G and van Tooren MJL. Enabling distributed multi-disciplinary design of complex products: A knowledge based engineering approach. *Journal of Design Research*, 5 (2007) 333-352.

[13] Bohnke D, Nagel B and Gollnick V. An approach to multi-fidelity in conceptual aircraft design in a distributed design environment. in: *IEEE Aerospace Conference*, 2011.

[14] Langen THM. Development of a conceptual design tool for conventional and boxwing aircraft. in, TU Delft, Delft, 2011.

[15] Brouwers YHA. Development of a kbe applications to support the conceptual design of passenger aircraft fuselages. in: *Faculty of Aerospace Engineering*, TU Delft, 2011.

[16] Vos R and van Dommelen J. A conceptual design and optimization method for blended-wing-body aircraft. in: AIAA (Ed.) *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Honolulu, Hawaii, 2012.

[17] Torenbeek E. *Synthesis of subsonic airplane design.* Springer, Delft, 1982.

[18] Raymer DP. *Aircraft design: A conceptual approach.* 4th ed., Washington D.C., 2006.

[19] Cooper DJ and La Rocca G. Knowledge-based techniques for developing engineering applications in the 21st century. in: *7th AIAA Aviation Technology, Integration and Operations Conference*, AIAA, Belfast, Northern Ireland, 2007.

[20] Kulfan B. Universal parametric geometry representation method *Journal of Aircraft*, 45 (2008) 142-158.

[21] La Rocca G. Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. *Advanced Engineering Informatics*, 26 (2012) 159-179.

[22] La Rocca G and van Tooren MJL. Knowledge-based engineering approach to support aircraft multidisciplinary design and optimization. *Journal of Aircraft*, 46 (2009).

[23] La Rocca G and van Tooren MJL. Knowledge-based engineering to support aircraft multidisciplinary design and optimization. *Proceedings of the Institution of Mechanical Engineering, Part G: Journal of Aerospace Engineering*, 224 (2010) 1041-1055.

[24] La Rocca G and Van Tooren MJL. Knowledge based engineering to support aircraft multidisciplinary design and optimisation. in: *ICAS 2008*, Anchorage, Alaska, 2008.

[25] van den Berg T. Parametric modeling and aerodynamic analysis of multi-element wing configurations. in: *Faculty of Aerospace Engineering*, TU Delft, 2009.

[26] Koning J, H. Development of a kbe application to support aerodynamic design and analysis. Towards a next-generation multi-model generator. in: *Faculty of Aerospace Engineering*, TU Delft, 2010.

[27] van Hoek M. Structural design, analysis and optimization of a lifting surface in a knowledge based engineering environment. in: *Faculty of Aerospace Engineering*, TU Delft, 2010.

[28] Wang H and La Rocca G. Knowledge based engineering support for fuselage load development. in: *28th International Conference of the Aeronautical Sciences*, Brisbane, AU, 2012.

[29] Berends J, van Tooren MJL and Schut JE. Design and implementation of a new generation multi-agent task environment framework. in: AIAA (Ed.) *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA, Schaumburg, IL, 2008.

[30] Nagel B, Bohnke D, Gollnick V, Schmollgruber P, Alonso JJ, Rizzi A and La Rocca G. Communication in aircraft design: Can we establish a common language? in: *28th International Conference of the Aeronautical Sciences*, Brisbane, AU, 2012.

[31] Carty A and Davies C. Fusion of aircraft synthesis and computer aided design in: *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, Albany, New York, 2004

## Copyright Statement