PARAMETRIC AIRCRAFT CONCEPTUAL DESIGN SPACE

Ingo Staack*, Raghu Chaitanya.M.V*, Patrick Berry¤, Tomas Melin*, Kristian Amadori¤, Christopher Jouannet¤, David Lundström*, Petter Krus*
**Linköping University, SE-58183 Linköping, Sweden*
¤Saab Aeronautics, SE-581 88 Linköping, Sweden
ingo.staack@liu.se

Keywords: *aircraft conceptual design, parametric modeling, sizing, XML database*

Abstract

This paper presents the development of a design framework for the initial conceptual design phase. The focus in this project is on a flexible database in XML format, together with close integration of automated CAD, and other tools, which allows the developed geometry to be used directly in the subsequent preliminary design phase. The database and the geometry are also described and an overview is given of included tools like aerodynamic analysis and weight estimation.

1 Introduction

As part of Saab's 75th anniversary celebrations in Linköping in June 2012, a formation flight showed the development of Swedish jet fighters over the last 60 years (see Fig. 1). Re-evaluation of these aircraft–with respect to the technology level of the time–and comparison with the practical experience gained during service gave engineers valuable information which could be applied directly in the following model.

Nowadays, aircraft design engineers struggle with the absence of "lessons learned" from previous projects due to the dramatically extended product life cycle and development lead times, as well as a never before seen complexity due to enlarged system integration. Conceptual aircraft design is also at the break-point between statistical/empirical methods and physical-related system calculations in order to enhance prediction accuracy. Multidisciplinary, holistic design is the solution and is becoming an additional field in the area of unmanned aircraft systems (UAS).

Fig. 1: Flying display of Saab fighters: (from left to right) J 35 Draken, J/S 29 Tunnan, JAS 39 Gripen, A/J/S 32 Lansen, 105 "SK60" and JA 37 Viggen

The trend towards multinational consortium based product development, like the European Eurofighter "Typhoon" project, has a further adverse effect: companies' functioning as aircraft component suppliers and system developers requires even sharper design engineers in order to secure the knowledge otherwise gained through their in-house design, development and construction. As a consequence, concept evaluation engineers have to take account of manned and unmanned aircraft, both military and civil, in fixed or rotary wing design. This requires a flexible, versatile and powerful framework during the conceptual aircraft design definition and evaluation phase, which should also support data sharing in collaborative research and industrial projects.

With the target of enhancing methodology and tool development, conceptual aircraft design has been a research field in the Swedish National Aviation Engineering Program (NFFP) since 1996 [1] and the most recent program framework development in this sector will be explained in this paper.

1.1 Related Work

In the field of aircraft conceptual design, a great many programs from research institutes and universities and also commercial products can be found. Some, like the RDS software from D. Raymer [2] are a direct implementation from classical aircraft design handbook methods, as described for example in [3], [4] or [5]. Here follows a short list of related university/research projects and programs:

- CEASIOM [6]
- padLAB : Preliminary Aircraft Design Lab [7]
- Vehicle sketch pad [8]
- Bauhaus Luftfahrt: Conceptual Design Tool (CDT) [9].

2 Conceptual Aircraft Design Methodology

The process in conceptual design development is focused on development and evaluation/comparison of different designs in order to benchmark the design and give feedback regarding the (partly negotiable) requirements. The main intention of the conceptual design phase is to reduce the number of possible layouts based on the design evaluation. The results of the conceptual phase should also:

- include all data and information used to develop the aircraft layout
- allow a backtrack of the requirements
- support flexible output routines in order to reuse this data directly in the tools of the subsequent preliminary design phase.

The last point takes primary aim at reuse of the CAD data but also means a tool-specific export for, for example, subsystem simulation and development tools. As a consequence, data handling and a flexible and fast implementable data interface emerges as the critical point in multidisciplinary work involving different tools with the same data setup [10]. In order to maintain flexibility, the database should also include as much functionality as possible.

2.1 One-Tool Concept

In the aviation industry, the introduction of full computer aided design (CAD) data based product development and production has led to a trend towards a one-tool strategy that includes multidisciplinary work. In the European aviation industry in particular, with CATIA [11] as a standard the embedment of the simulation program Dymola [12], based on the Modelica language, into the CATIA V6 environment, has establish a new holistic approach together with the ABAQUS finite element method tool already included in CATIA V5. These programs are linked under the umbrella of CATIA V6 by means of a proprietary data format.

2.2 One Database Concept

By contrast, research institutes and universities, who are more interested in maximum flexibility and tool integration (in order to enlarge multidisciplinarity), prefer a non-proprietary (open source) database definition. One recently published proposed standard is the Common Parametric Aircraft Configuration Scheme (CPACS) devised by the German Deutsche Zentrum für Luft- und Raumfahrt (*DLR*) [13][14]. This data setup has been tested and used for several years at DLR in preliminary projects with different focus, for example aircraft noise emissions, structure analysis or even whole fleet simulations. [15] describes the implementation of CPACS within the CEASIOM project to improve the data definition. One drawback of using CPACS directly in aircraft conceptual design may be the extent of the data description, including fleet, helicopters and land-based vehicles up to a very highly detailed level of geometry and structure description. The approach in this project was also to define the data setup in a very robust, easy manner, accepting the adverse effect of design space limitations on the advantage of reduced complexity. Furthermore, the data definition should represent the mindset (of the developer) and represent the main design parameters directly. It must be remembered that during initial data setup development, the CPACS format definition was not available to the authors of this paper, but the similarities in data definition between CPACS and the small subset in this project–which had already been defined when the CPACS format was published–were remarkable. As regards design details (close to or already in preliminary design), CPACS terminology has partly been directly adapted.

The main disadvantage of a central data setup definition is the size of the data file size, together with failure to update (and manage updated information), in particular when parameters are modified in different tools. These problems mainly occur when applying different analysis methods on the data setup.

With the complexity and multidisciplinary nature of conceptual design, it might also be useful to use high-level, graph-based design (modeling) languages to model the product data, including requirements and system configuration and architecture. One example is the Unified Modeling Language (UML) which is shown in [16] where it is applied to satellite design. SySML, as a development of UML, might be even more appropriate.

2.3 Challenging the Gap between Design Definition and Evaluation

Alongside the data setup definition topic, the matter of retrieval of the “perfect” designing tool appears. Somewhat oversimplified, this tool should support the developer in creating designs out of the requirements and evaluating/ benchmarking these designs according against them.

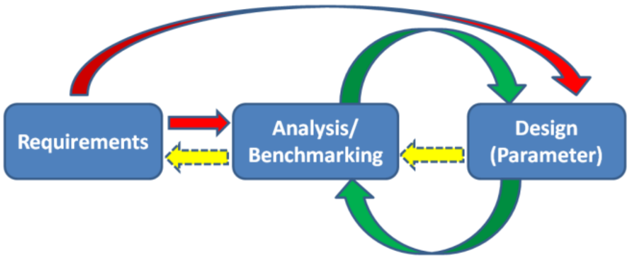


Fig. 2: Conceptual aircraft design phase: Requirement influence (red), main design loop (green) and requirement update (yellow) correlation

In reality, finding the right design will be an optimization loop between the design definition that is being evaluated and the requirements, which also involves negotiation and balancing of the requirements (see Fig. 2). This is particularly important in the conceptual phase.

Whereas for design definition a (subset of a) CAD program is particularly useful, it does not normally fit concept analysis and evaluation well. In the latter case, a scripting language combined with an adapted graphical user interface (GUI) is much more usable and gives the developer the possibility to add benchmarking or optimization algorithms of his or her own. Based on these considerations, CATIA V5 and Matlab were chosen as the main tools for the design process.

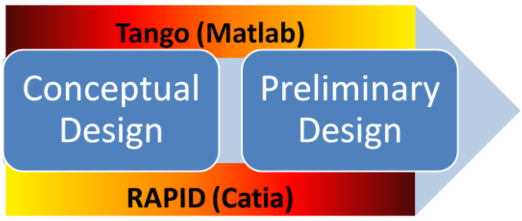


Fig. 3: Parallel implementation concept with a step-less fadeout of Matlab based design definition in more detailed design

In order to maintain flexibility and allow the developer to choose his or her preferred work method, both programs should be implemented in parallel (see Fig. 3). Switching between the two should be possible at any time. In normal mode, the common database is therefore hidden in the background by tool-specific XML interfaces so that the user feels that communication in both applications is done directly. Due to the nature of the work with these programs, and especially the limitation of graphical representation outside a full CAD environment, it is recommended that detail design and structure definition is more related to CATIA than to Matlab.

Combining an interpreter language with a graphical user interface (GUI) and CAD software, in this case both Matlab and CATIA V5, is not a new idea; it has already been used, but mainly for preliminary design, e.g. in the form of "PadLAB" (Preliminary Aircraft Design Lab), by the Institute for Aerospace (ILR), Technische University Berlin with the focus on cabin layout [7].

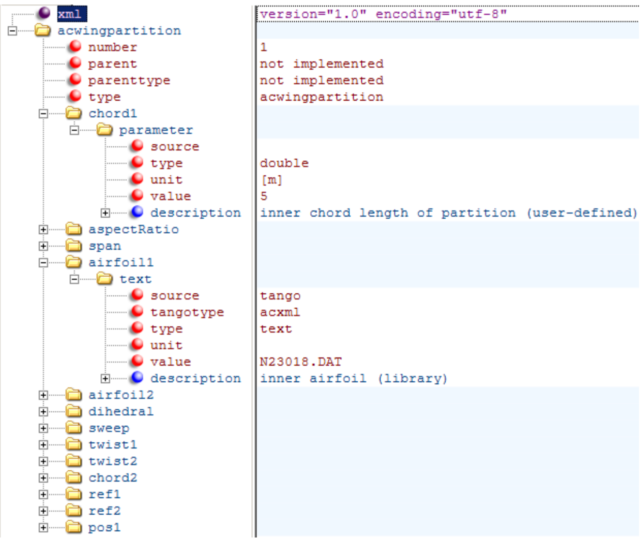
3 Implementation of the Conceptual Aircraft Design Framework

In the following, the data setup and the two programs Tango (Matlab based) and RAPID (CATIA V5 based) will be explained in detail.

3.1 Database Setup and Handling

Based on the requirements of a general database with a focus on flexible access, XML representation was chosen. This representation can be accessed by practically any programming languages via standard interfaces like the Document Object Model (DOM). Furthermore, by defining the structure as an XML Style Sheet (XSD), data setups can easily be checked for validity. Transitions within the parsers between the XML data setup and each connected tool can be implemented fairly quickly and easily with the help of the XML Stylesheet Language for Translation (XSLT). These translation files are implemented in the XPath language [18] to access nodes or node-sets in XML documents. This language however, has limitations regarding mathematical operations and complex parsing actions but functions well for hierarchical level translations.

During database style development, a particular focus was laid on a robust, parametric based data description in order to allow for automation and the application of optimization algorithm later on. The data setup should also be as similar as possible, regardless of aircraft type (civil/military or UAV). Therefore, and because the focus is on conceptual and not preliminary design, design limitations due to too strict data definition have to be accepted. As a consequence, this database might not function well for programs creating generic design apart from the (pre)defined design space.



```
xml                      version="1.0" encoding="utf-8"
acwingpartition
    number               1
    parent               not implemented
    parenttype           not implemented
    type                 acwingpartition
    chord1
        parameter
            source
            type         double
            unit         [m]
            value        5
            description  inner chord length of partition (user-defined)
    aspectRatio
    span
    airfoil1
        text
            source       tango
            tangotype    acxml
            type         text
            unit
            value        N23018.DAT
            description  inner airfoil (library)
    airfoil2
    dihedral
    sweep
    twist1
    twist2
    chord2
    ref1
    ref2
    pos1
```

Fig. 4: Database example of a wing section definition

One example, shown in Fig. 4, is the definition of a wing section, based on the cross-section definition of airfoils. All airfoils are consistently defined by a Bézier curves method presented in [19].

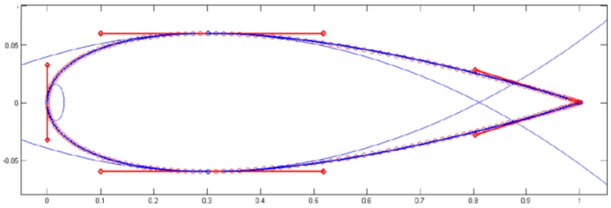
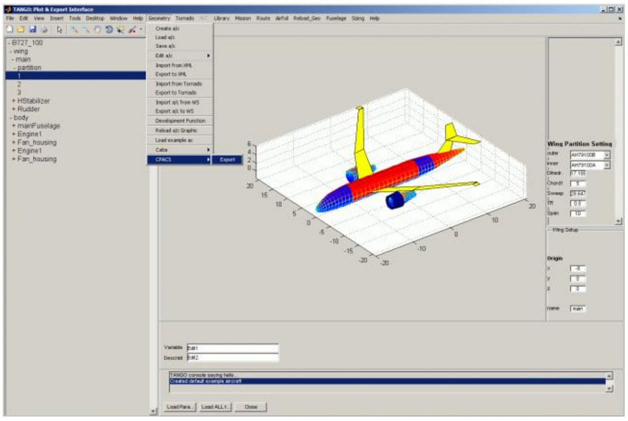


Fig. 5: Airfoil definition representation by Bézier curves control points

This method uniforms different airfoil descriptions in a very robust, computer-interpretable data definition, based on only a few dominant parameter (see Fig. 5). This definition has nevertheless a wide design space with the only drawback that it is not possible to model airfoils with an S-shaped trailing edge.

3.2 Tango (Matlab) Implementation

The Tango implementation makes use of the new class definition features of the Matlab language [20]. This enables an object oriented programming (OOP) implementation close to C++ together with interpreter language benefits regarding debugging (fast development) and the useful console input capability. The drawback is the vastly slower calculation speed and perhaps stability problems when program size and complexity increase compared to C++.
However, using Matlab was a requirement from the industrial partner, because most of their engineers are used to this language. However, considering the open source approach, Python would be a proper alternative.



outer: AH79100B
inner: AH79100A
Chord1: 5
TR: 0.5
Span: 10
x: -8
y: 0
z: 0
name: main
Variable: Edit1
Descript: Edit2

```
TANGO console saying hello...
Created default example aircraft
```

Fig. 6: Screenshot of the Tango start-up window

The strictly class based implementation allows for a modular program implementation based on physical systems, components and functions. All XML database support and all calculations specific to components, systems or parts (e.g. weight estimation) are performed within the class. This also allows for fast component exchange and replacement by means of standardized coupling ports and functions and ensures definition consistency through direct parameter access restrictions.

A GUI with a rather strict hierarchy, supporting an easy to use working environment with a graphical representation of the current topic, is implemented on top of the class setup. Alongside this mode, design development and evaluation can be performed directly from the Matlab console or by using a certain scripting language to address the classes. Fig. 6 shows the default start-up GUI window and Fig. 7 a small script snippet of an aircraft geometry definition.

```
ac = acdata('exampleAC');

ac.setting.setProperties('cg', [5.1807, 0,0.0972],...
                         'rp', [5.6223 0 0.0972]);
% 1.)Main wing
ac.addwing('main');
ac.wing{1}.setOrigin(-8, 0, 0);
ac.wing{1}.addPartition();

%Inner wing:
ac.wing{1}.partitions{1}.setProperties('chord1',5,...
                   'span',      10, ...
                   'taper',     0.5, ...
                   'sweep',     0.5, ...
                   'dihedral',  0.3, ...      %pi/4
                   'twist1',   0.1, 'twist2', -0.1 );
%Outer wing:
ac.wing{1}.addPartition();
ac.wing{1}.partitions{2}.setProperties('sweep', 0.7);
%Winglet:
ac.wing{1}.addPartition();
ac.wing{1}.partitions{3}.setProperties('span',1.0,...
                  'tr',   1.0, ...
                  'sweep', 0, ...
                  'dihedral', 0, ...
                  'twist1',    0.0, 'twist2', 0.0);
```

Fig. 7: Example of a basic aircraft geometry (wing) definition script

Besides the geometry (and system integration) definition, and sizing, weight estimation and aerodynamics are the central points during design development. Sizing is done classically by statistical methods as well as easy physics, supporting the user with the usual sizing diagrams, taking the selected certification (JAR, FAR 23/25) into account.

Fuselage weight is calculated sector-wise taking into account the airplane structure, location, size and shape of doors, windows, etc. as well as the installed (sub)system components. Here, the old classically weight calculation, based on the calculation of an equivalent skin thickness defined by the shear force and bending moment and adding penalty weight for windows, hatches, installations and (sub)system components [21], is combined with a more physically related weight estimation of the included systems and components. A detailed explanation of this method can be found in [22].
The wing is calculated either in the same way as the fuselage or from the wing structure CAD data, if already defined in detail.

Initial aerodynamic calculations–mainly needed for thrust and fuel consumption estimation–are calculated by the lattice vortex panel method program TORNADO [23]. The export to TORNADO and the calculations take place automatically, and hidden to the user, within few seconds but can be changed if needed. With the help of the parameters obtained, the sizing results and the engine data, classical mission performance is calculated.

For more accurate evaluation capability, a full aircraft simulation export capability including (sub)system integration is currently under development using the Hopsan simulation package developed at Linköping University. Full system simulation using this software is presented in [26]. The export functionality is relatively generic and models can also be exported to Modelica. Besides higher accuracy, the simulation will give the designer direct feedback and thereby a better understanding of the design as well as the potential to investigate the effects of different (sub)system architectures and system integration. This topic becomes especially necessary with the transition to cross-linked electrically driven systems like the environmental control system (ECS) or the anti-ice system, which have a significant impact on both aircraft operating empty weight (OWE) and energy efficiency.

Other methods for design investigation and evaluation include:

- sensitivity and robustness analysis according to [24]
- maneuver analysis (especially for military aircrafts).

Currently implemented export and import capabilities are:

- TORNADO
- CPACS format (limited)
- CEASIOM XML format [25]
- Saab in-house aircraft conceptual design tool.

Most of these translations are solved by using XSLT translation style sheets.

Central parts in Tango, in contrast to RAPID, are the functional onboard power system implementation and the control topology. Currently implemented systems are:

- **Propulsion system:** Based on performance lookup tables normally supplied by the original manufacturer. Reference thrust, bypass ratio, pressure ratio and maximum turbine inlet temperature, for example, are generically derivable out of the central design parameter. Propeller-piston configuration is not implemented yet but may become more important with the upcoming new Kerosene/Diesel (piston) engines on the UAV market.
- **Primary Flight Control System** (PFCS): Includes control surface geometry, control topology and (hydraulic) actuator power system. This section–together with the hydraulic system description–is taken directly from the CPACS format [13]
- **Landing gear system,** including track animation and actuator system
- **Environmental Control System:** Cooling and pressurization of the aircraft.

All these system classes include configuration help and data libraries in order to support the developer with automatically default adapted (architectures and sizing) systems.

3.3 RAPID (CATIA) Implementation

RAPID (Robust Aircraft Parametric Interactive Design) is a knowledge-based Aircraft Conceptual Design tool built in CATIA.

The main motivation to use CATIA for this purpose is to enable the propagation of the design and its contents from conceptual to preliminary design. The surfaces generated are A-class surfaces and can thus be used directly for initial aerodynamic analysis. It makes use of the two powerful automation technologies embedded in CATIA, viz. Visual Basic (VB) scripts and Knowledge Patten (KP). Power Copies (PCs) and User Defined Features (UDFs) are created and utilized by the VB scripts and KP respectively for instantiation.

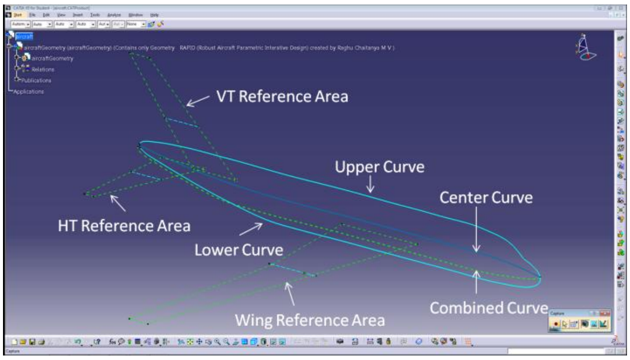


Fig. 8: Default RAPID start-up window

The design in RAPID is achieved either by a bottom-up approach or by modification of an existing aircraft configuration example. The flexibility of the model implementation allows changing from a civil aircraft to fighter or UAV. In a bottom-up approach, the user begins by modifying the fuselage according to requirements and later modifying the wing. From here on the other lifting surfaces are automatically sized; only configuration and positions have to be defined manually. Once the default RAPID aircraft model is loaded into CATIA, different aircraft from the XML database can be loaded and the model updated (see Fig 8 and Fig. 9).

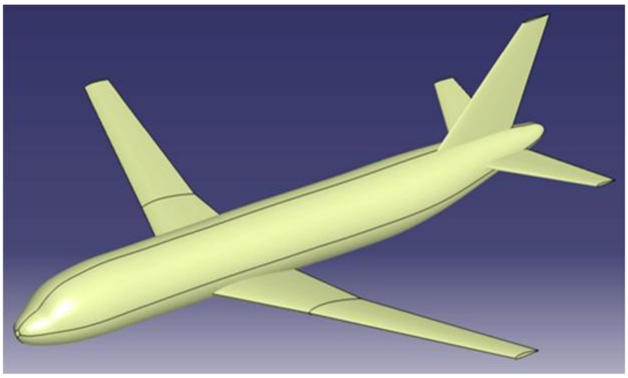


Fig. 9: Civil transport aircraft geometry loaded from database

The detailed Geometric Model (GM) can be sent directly to aerodynamic analysis and, as the model has fewer surfaces, it can be meshed with less effort and the process may be automated in the future. The GM is the basis for the Structural Model (SM) and the number of spars and ribs can be chosen for the lifting surfaces as well as frames and stringers for the fuselage. The SM can be meshed automatically and is prepared for initial structure analysis. Heading towards preliminary design, control surfaces, windshield, fairings and winglets etc. can be defined for both the GM and the SM. Pilot model, cockpit model and cabin layout will be features added in the future.

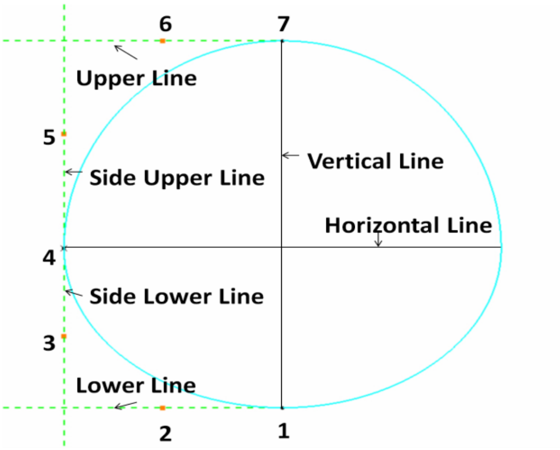


Fig. 10: Fuselage cross-section definition by 3rd-order Bézier curves

The data exchange to and from the XML database is implemented using VB scripts. All lifting surfaces make use of the same airfoil definition, shown in Fig. 5. A similar approach is used for the fuselage, shown in Fig. 10; here, a third-order Bézier curve is used to describe a quarter section of the fuselage cross-section where the upper and lower lines measure an angle with respect to the horizontal line, while side upper line and side lower line measure an angle with respect to the vertical line. Points 2, 4 and 7 are the intersection points with the fuselage curves shown in Fig. 10. Points 2, 3, 5 and 6 move along the respective curves and are positioned as ratios.

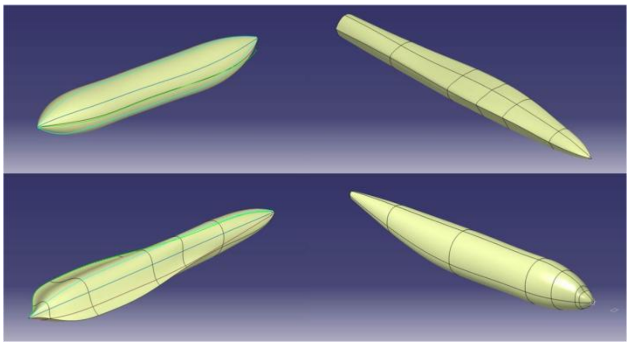


Fig. 11: Examples of different fuselage layouts

Fig. 11 shows some examples of both civil and military fuselage layouts that can be modeled.

Engine sizing is an additional feature of RAPID. Turbofan and turbojet engines can be sized depending on design parameters (e.g. the bypass ratio). The nacelle is designed from the size of the engine; mixed flow and separate jet nacelles can be chosen accordingly. A wide range of parameters can be changed for the nacelle, from inlet diameter to exhaust diameter. Different gear boxes and pylon types can also be modeled. The pylon design depends on the type of nacelle; start and end positions can be chosen as necessary. Two configurations of the engine and engine installation are shown in Fig. 12.

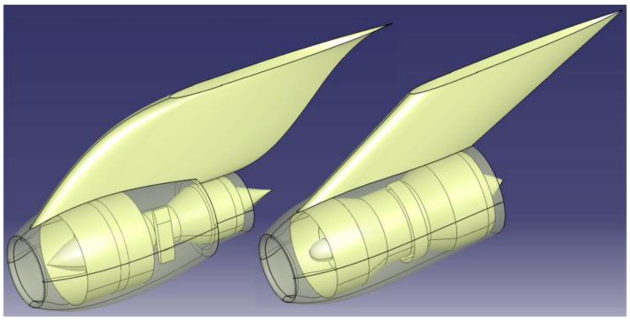


Fig. 12: Turbofan with trapezoidal gearbox, separate jet nacelle and smooth leading/trailing edge nacelle (left) and turbojet engine with circular gearbox, mixed flow nacelle and straight leading/trailing edge pylon (right)

Military air inlet channel definition work is ongoing, but no satisfying solution for this problem has been bound yet.

4 Discussion

According to the framework concept motivated in Chapter 2, the focus was on a robust, parametric data definition. This database supports extensive data export and import capabilities for flexible tool integration. The closeness of the CAD implementation to the data setup simplifies the integration of geometry-related sophisticated evaluation methods such as structure (FEM) and aerodynamic (CFD) analysis.

One central issue in all modeling tools is to create a constraint design space where the user should have as much freedom as possible, while designs that are clearly not valid are not present in the design space. This minimizes the amount of information that needs to be provided by the user. With the tools provided in this framework, the user can quickly generate detailed concepts with a minimum of parameters. Another improvement regarding old-fashioned implemented approaches (e.g. Fortran-based sizing tools) is the direct feedback to the developer, either directly in the CAD or in the Tango GUI. The direct graphical feedback (“what you see is what you get”) is useful to avoid wrong inputs and unburden the user of imagining the design on his or her own.

During the project, the limitations of the used software/languages became quite clear; that is, on the one hand, the limited geometry definition and graphical representation capability outside a complete CAD environment, and on the other hand, the extremely slow code execution speed in a CAD environment (here VB scripts in CATIA). These experiences back the initial decision to use two main tools, Matlab and CATIA, in order to balance the needs for both, design definition and evaluation.

The replacement of empirical methods by physical implementation of (onboard power) systems and equipment is an indirect benefit of the above mentioned topics: through the extended usage of XML based airfoil-, wing-, aircraft-, sub- and system-component libraries and configuration help as well as the user-friendly machine-human interface, the modeling lead time can be shortened. This allows additional design properties, especially onboard power systems to be defined with the same time effort as before. This increase in (design) information can lead to higher estimation precision than empirical formulas can**.**

Drawbacks in this design framework are the absence of requirement handling and the rudimentary (product life cycle) cost analysis. The cost analysis can actually be seen as the main benchmark requirement in the early conceptual design phase, with the focus on feasible studies for requirement definition. This topic might be better solved with the competition approach mentioned, creating automated design with the help of graph-based design languages. However, due to ongoing implementation work the framework has not been tested in case studies, re-evaluation of existing aircraft or by comparing the results with real data, handbook methods and other conceptual design tools.

4.1 Outlook

Negotiations regarding software and code publication are ongoing, but from Linköping University's point of view, the framework should be made available as open source software in order to support cross-company and cross-university collaborations and knowledge sharing.

The main focus in a continuation of this project will be on:

- simulation integration
- additional and refined subsystem integrations
- enhanced weight estimation methods
- methods for requirement handling (implementation)
- tool verification and adaptation through aircraft re-evaluation
- adding/enlarging databases/libraries
- adding tracking of the "designer workflow" (how has the designer built up the model) in order to replay project-specific aircraft design development for both, education/training purpose as well as within requirement adoptions and optimization loops.

5 Conclusion

A flexible conceptual aircraft design framework, based on a solid data setup has been developed in collaboration between Linköping University and Saab Aeronautics. The robust parametric data definition together with the parallel/matching CAD model enables the application of optimization algorithms and a direct data reuse in the subsequent preliminary design phase. The split-up into a stand-alone, XML-based database and the design definition and analysis tools enable a flexible integration of external tools. Additionally, this database supports the geometry definition process through extensive component libraries and configuration functionality.

Acknowledgement

Funding of this work was provided by NFFP, the Swedish National Aviation Engineering Program. The contributing authors wish to thank the NFFP founders for this support.

References

[1] VINNOVA. Swedish national aviation engineering research programme, http://www.vinnova.se/en/Our-activities/Cooperation-Programmes/National-Aviation-Engineering-Research-Programme/ (June 27, 2012)

[2] Raymer D. *RDS-student: software for aircraft design, sizing, and performance*, Volume 10, AIAA educations series, Washington DC, 2006

[3] Raymer D. Aircraft design - a conceptual approach, 3rd edition, AIAA educations series, Washington DC, U.S.A, 1999

[4] Roskam J. *Airplane design- part1: Preliminary Sizing of Airplane*, DARcoporation, Lawrence, 1985

[5] Torenbeek E. *Synthesis of subsonic airplane design*, Delft University Press, Delft, Netherlands, 1995

[6] CEASIOM. Computerised Environment for Aircraft Synthesis and Integrated Optimisation Methods software, http://www.ceasiom.com (June 27, 2012)

[7] PADLab software, http://www.luftbau.tu-berlin.de/menue/forschung/padlab/ (June 27, 2012)

[8] Hahn A. Vehicle Sketch Pad: Parametric geometry for conceptual aircraft design, *Proc 48th AIAA Aerospace Sciences Meeting*, Orlando, Florida, 2010

[9] Ziemer S. A conceptual design tool for multi-disciplinary aircraft design, *Proc Aerospace Conference*, IEEE, Big Sky, *Montana*, USA, 2011

[10] Johansson B, Jouannet C, Krus P. Distributed aircraft analysis using web service technology, SAE Transactions, American Technical Publishers Ltd,Vol 112, Part 1, pp 445-453, Orland, U.S.A., 2003

[11] Dassault Systems. *CATIA software Version V5-6R2012*, http://www.3ds.com/products/catia/portfolio/catia-v5/latest-release/ (June 27, 2012)

[12] Modelon AB. *Dymola software*, http://www.modelon.com/products/dymola/ (June 27, 2012)

[13] DLR. *CPACS software*, http://software.dlr.de/p/cpacs/home/ (June 27, 2012)

[14] Liersch C, Böhnke D. A unified approach for multidisciplinary preliminary aircraft design, *Proc 2nd CEAS European Air and Space Conference*, Manchester, UK, 2009

[15] Rizzi A, Zhang M, Nagel B, et al. Towards a unified framework using CPACS for geometry management in aircraft design, *Proc 50th AIAA Aerospace Sciences Meeting and Exposition, Nashville, U.S.A, 2007*

[16] Groß J, Rudolph S. Generating simulation models from UML - A FireSat example, *Proc DEVS Integrative M&S Symposium*, Orlando, FL, USA, 2012

[17] Arnold P, Rudolph S. Bridging the gap between product design and product manufacturing by means of graph-based design languages, *Proc 9th Tools and Methods of Competitive Engineering*, Karlsruhe, Germany, 2012

[18] World Wide Web Consortium (W3C) XML. *XML Path Language (XPath), Version 2.0*, http://www.w3.org/TR/xpath20/ (June 27, 2012)

[19] Melin T, Amadori K, Krus P. Parametric wing profile description for conceptual design, *Proc CEAS European Air and Space Conference*, Venice, Italy, 2011

[20] Mathworks. *Matlab software: MATLAB Classes and Object-Oriented Programming*, R2012a, http://www.mathworks.se/help/techdoc/matlab_oop/ug_intropage.html (June 27, 2012)

[21] Hammitt R. Structural weight estimation by the weight penalty concept for preliminary design, *Proc 15th National Conference of the Society of Aeronautical Weight Engineers*, San Diego, California, U.S.A., 1956

[22] Patrick B, Jouannet C, Krus P. Recycling old weight assessment methods and giving them new life in aircraft conceptual design, *Proc 28th Congress of the International Council of the Aeronautical Science*, Brisbane, Australia, 2012

[23] Melin T. *TORNADO a Vortex-Lattice Matlab implementation for linear aerodynamic wing applications*, Master Thesis, Royal Institute of Technology (KTH), Department of Aeronautics, Sweden, 2000

[24] Krus P. Computational tools for aircraft system analysis and optimization. *Proc 26th International Congress of the Aeronautical Sciences*, Anchorage, Alaska, USA, 2008

[25] Puelles A, et al. *CEASIOM XML file definition*, Version 1.0, Manual, http://www.ceasiom.com/downloads/CEASIOM%20Documentation/ceasiom-xmlfiledefinition.pdf (June 27, 2012)

[26] Krus P, Braun R, Nordin P, et al. Aircraft system simulation for preliminary design. *Proc 28th Congress of the International Council of the Aeronautical Science*, Brisbane, Australia, 2012

Copyright Statement