Implementation of a Heterogeneous, Variable-Fidelity Framework for Flight Mechanics Analysis in Preliminary Aircraft Design

T. Pfeiffer[1], B. Nagel[1], D. Böhnke[1], A.Rizzi[2], M. Voskuijl[3]

[1]German Aerospace Center (DLR), Air Transportation Systems (LY)
Blohmstraße 18, 21079 Hamburg, Germany
[2]KTH Royal Institute of Technology (KTH)
10044 Stockholm, Sweden
[3]Delft University of Technology, Faculty of Aerospace Engineering
Kluyverweg 1, 2629 HS Delft, the Netherlands

This paper presents the implementation of a framework for flight mechanical analysis with adaptable aerodynamics methods in the preliminary stage of aircraft design. Basis to the framework are software modules for the disciplines conceptual design, flight mechanics and aerodynamics which have been developed within different design frameworks and which contain heterogeneous interfaces. The focus is placed on the technique for the integration of the different analysis components and the realization of the variable-fidelity capability. Backbone of the framework is the XML-based 'Common Parametric Aircraft Configuration Schema' (CPACS) which was extended in course of this study. The paper introduces the utilized analysis codes and software modules, presents their integration into the framework and discusses the faced challenges and developed solutions. Exemplary results are shown as illustrative use cases. Analogies of linked design systems lead to a standardized integration process. In a next step the homogenization of different design systems by a unified parameterization appears to be feasible.

1. INTRODUCTION

Distributed frameworks for multidisciplinary aircraft analysis and optimization are currently becoming state-of-the-art. Tough multiple commercial and open source software products are available today which support various tasks required, the implementation of an over-all aircraft optimization framework causes significant effort. A major problem is constituted by linking and adapting the large number of disciplinary analysis tools necessary. The common practice today is that in each framework project an own integration concept is developed. Once conventions for the integration have been agreed and implemented within one framework, it becomes very difficult to introduce changes to the interface concept. Thus, each framework internally offers homogeneous interface and tools - but each framework is solitary and not directly compatible to other frameworks.

High fidelity measures are used today already in the early stages of design. With this trend, also a more collaborative way of working is becoming state-of-the-art in order to share the workload. Each time a consortium is formed it is necessary to set up a new analysis framework. Thus; the re-usability of framework components in different environments has to be considered as a field of research with a paramount impact on the efficiency of development and research.

This study investigates the implementation of a distributed analysis framework focused on flight mechanics making use of heterogeneous analysis modules. In a first step a framework is realized from components sourced from different existing aircraft design frameworks. The second step involves different fidelity tools for aerodynamics. The final conclusions focus rather on the experiences made in the integration task and potential solutions for more efficient collaboration techniques than the results from the aircraft design task which are shown as necessary use case.

2. STATE OF THE ART AIRCRAFT DESIGN FRAMEWORKS

This chapter introduces three frameworks for multidisciplinary aircraft design which resulted from medium sized research projects like of the European Framework Programmes level one type. They are representatives of the state-of-the-art and are the sources of the heterogeneous framework components which will be discussed in the later chapters.

2.1. Design and Engineering Engine

Since 2002, TU Delft has started to develop a *Design and Engineering Engine* (DEE) with the purpose of supporting and accelerating Multi Disciplinary Optimisation (MDO) of complex products through the automation of the non-creative and repetitive design activities. Further efforts were made to widen the scope for creative design using *Knowledge Based Engineering* (KBE) principles to find a cost-effective and innovative design. The first development activities of DEE started within the European MOB project (Multidisciplinary Optimisation of a Blended Wing Body), which launched in the year 2000 for a duration of 3 years

[18] [19] [22]. The DEE consists of a multidisciplinary collection of design and analysis modules and is able to automatically interface and exchange data and information between them. Each module is loosely coupled and can vary in number and type according to the design case at hand. Distributed system functionality allows the implementation of modules form different computers. The DEE includes a Multi-Model Generator (MMG) which is the central geometry model generator and automatically generates input data and specific disciplinary models for various disciplinary analysis tools. Examples of implemented disciplines were aerodynamics and structures with different levels of fidelity.[18]

2.2. Computerised Environment for Aircraft Synthesis and Integrated Optimisation Methods

The 'Computerised Environment for Aircraft Synthesis and Integrated Optimisation Methods' (CEASIOM) is a software suite that contains disciplinary tools for conceptual and preliminary aircraft design. It is developed to support engineers in the conceptual design process with emphasis on the improved prediction of stability and control properties through the use of higher-fidelity CFD methods. CEASIOM was developed within the frame of the SimSAC project (Simulating Aircraft Stability And Control Characteristics for Use in Conceptual Design) funded by the European Commission's 6th Framework Programme on Research, Technological Development and Demonstration. CEASIOM integrates the principal design disciplines aerodynamics, structures, and flight dynamics (FIGURE 1). The aerodynamic discipline includes different fidelity modules. It includes TORNADO, a vortex lattice code for low-speed aerodynamics, and EDGE, a CFD code solving the Euler or RANS (Reynolds Averaged Navier-Stokes) equations hence is appropriate also for the transonic regime and for extremes of the flight envelope. CEASIOM uses an own developed XML based data exchange file, to communicate between each disciplines. The CEASIOM xml definition is documented in [8]. That xml exchange file is constraint on the amount of data. For example, only three wing sections can be defined. [7] [9]
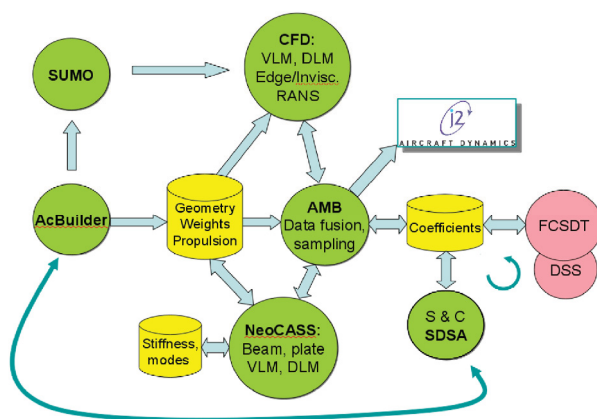

FIGURE 1. CEASIOM Framework

2.3. TIVA and VAMP

In 2005, the German Aerospace Center DLR started an internal project, called TIVA (Technology Integration for the Virtual Aircraft) and in 2009 a successor-project designated VAMP (Virtual Aircraft Multidisciplinary Analysis and Design Processes). The main goal was to further strengthen the multidisciplinary collaboration in the field of preliminary aircraft design. Ten institutes of the DLR have participated, each with a distinct disciplinary specialization and located at several sites in Germany. The environment allows collaboration with the disciplinary groups running their tools on own servers. Thus, the involved specialists operate their codes in their native environment and are integrated into the collaborative analysis process. The framework program ModelCenter [23] is used for linking the disciplinary capability codes into the framework. For the communication between the disciplines, a XML based exchange format called CPACS (Common Parametric Aircraft Configuration Schema) was developed which will described below. This exchange file allows the setup of tool chains with a high agility for running analysis and optimization studies. Various disciplines are involved in the current project: conceptual aircraft design methods, aerodynamics, structures, propulsion, flight performance, handling qualities, noise, cost modelling, etc. [17]. The system is currently becoming a standard at DLR and has been used in multiple projects with applications in the design of civil and military fixed wing aircraft, rotorcraft, gas turbines and the analysis of climate and acoustic environmental impact from mission level till the intermodal transport system level.

3. CPACS

In multidisciplinary environments, a central model approach delivers the benefit of decreasing the number of interfaces as illustrated in FIGURE 2. The interfaces, which are called wrappers, are set up for each analysis module to communicate with data format and translates between the global and the tool specific data format and vice versa.[4]
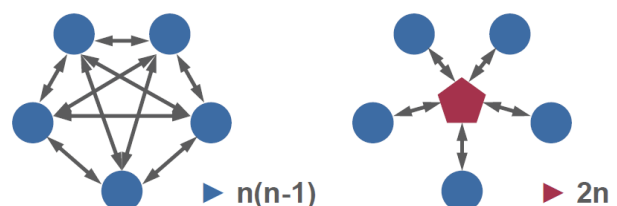

FIGURE 2. Central Data Model Approach

The Common Parametric Aircraft Configuration Schema (CPACS) is a highly adaptable data format for aircraft design. It includes the description of the aircraft configuration, tool-specific information to control the analysis process and analysis results. CPACS is used as backbone of the presented framework.

CPACS has a hierarchical 'tree-like' structure with one root element on the top level FIGURE 3. The structure is defined in a schema file (XML Schema Definition XDS) which includes the rules how elements, attributes, etc. may be included. The elements under the root level are a

header element, which includes basic information of the file, a vehicles element, which includes the definition of the aircraft geometry and characteristics, a missions element including information of mission segments, an airports element including location and runway information of airports, a fleets element, that includes information of air tracks and a tool specific element for process control information. The level depth is variable and dependent of the level of detail to be considered for the specific study. The aircraft geometry is described in CPACS/vehicles/aircraft/model (see FIGURE 3) including all geometry of the wings, fuselage, engines etc. In principle CPACS can be defined to hold the definition of an aircraft with an arbitrary level of fidelity and complexity. Wings, for example, can be described with several sections; they can split and joined, and can explicitly define details like ribs and spars till individual rivets if required.
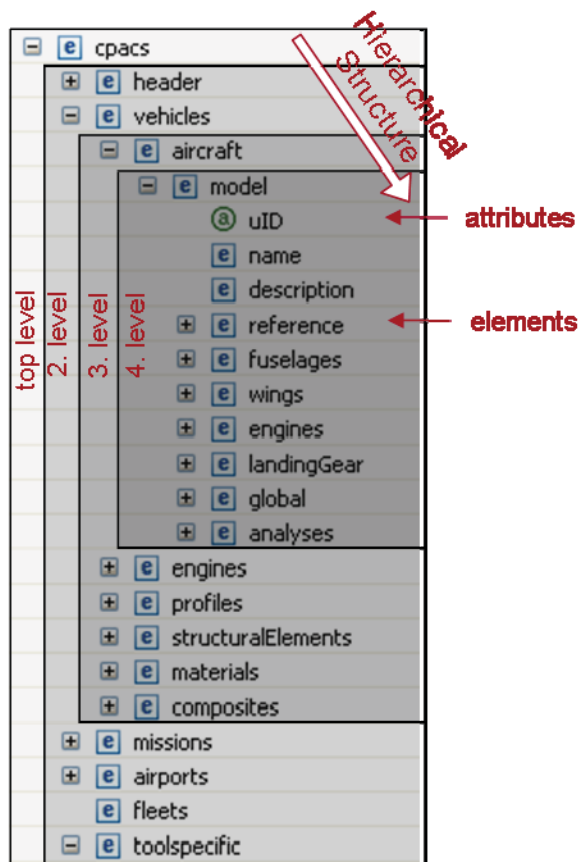

FIGURE 3. CPACS Structure

The handling of CPACS allows for including several methods for programming a wrapper. A wrapper includes a script which reads CPACS, translates the CPACS format into the required input format for the method. After the method has run, the method output format translates to CPACS format and writes back the CPACS as illustrated in FIGURE 4. Process control information for the method like tool settings are implemented under the CPACS tool specifics node.

Due to its flexibility and potential level of details, the CPACS language can become quite complex in some applications. This can cause to problems for inexperienced users. In principle it is necessary to validate the wrapped

method, to ensure that the wrapped model is identical. Complexity is necessary because of the needed flexibility to build up arbitrary aircraft geometries. There are rules which must be followed when using CPACS. All regulations for the CPACS dataset are summarized in the CPACS documentation.
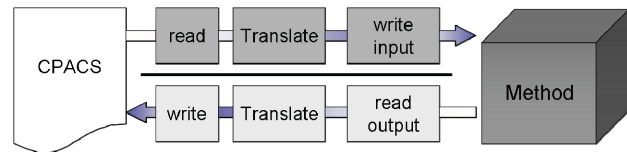

FIGURE 4. CPACS Wrapper

In general, CPACS allows all tools which have a wrapper to speak the same language. That is the basis of communication and collaboration of tools. So it is a valuable investigation to write wrappers to set up or extend a tools portfolio.

To support the handling with CPACS, two libraries for data manipulation are available. These are called TIXI (TIVA XML Interface) and TIGL (TIVA Geometry Library) and described in [20].

For using CPACS in Matlab there are two Matlab functions (xml2mat and mat2xml) for import (geometry, tool specifics data) and export (tool results) to the xml CPACS file.

4. BASIC HETEROGENEOUS FRAMEWORK

The aim of the current study was to investigate how analysis modules form different sources can be integrated into a CPACS-based framework. The analysis modules are provided with interfaces for their source framework and therefore have to be considered to be heterogeneous in the context of this new framework. Of particular interest were best practice methods for the integration. If also the source framework contains a central data format of a similar level of detail, a general bridge between the two formats can be established. If there is no central data format available or significantly different level of details require the translation into a higher or lower dimensional variable space, wrappers for the individual tools might become necessary. In both cases software programs called wrappers are necessary to trigger the execution of the integrated components and to translate between CPACS and the tools' individual namespaces.

An example framework for flight mechanical analysis in the context of preliminary aircraft design was set up in collaboration of DLR, TU Delft and KTH Stockholm (FIGURE 5). The realized analysis chain starts with a preliminary aircraft design and culminates in the capability to fly the aircraft in a desktop flight simulator. The Framework includes three disciplines. The initial aircraft configuration is designed using the conceptual aircraft design tool VAMPzero, from the DLR toolbox. A first aircraft model is then instantiated in the CPACS format. Alternatively, conceptual or preliminary aircraft designs can be loaded from a database. The subsequent analysis code TORNADO calculates aerodynamic moments and forces of the configuration in differed flight conditions. Finally, the flight mechanical behaviour is simulated and

analyzed using the Flight Mechanics Toolbox, developed at Delft University of Technology. This toolbox was used in the DEE framework. Besides fully automatic analysis, there is also the capability to interactively conduct piloted simulations using the open source desktop flight simulation software FlightGear. The main framework control applications are realized in Matlab. The disciplinary components and their implementation in the framework will be described below.
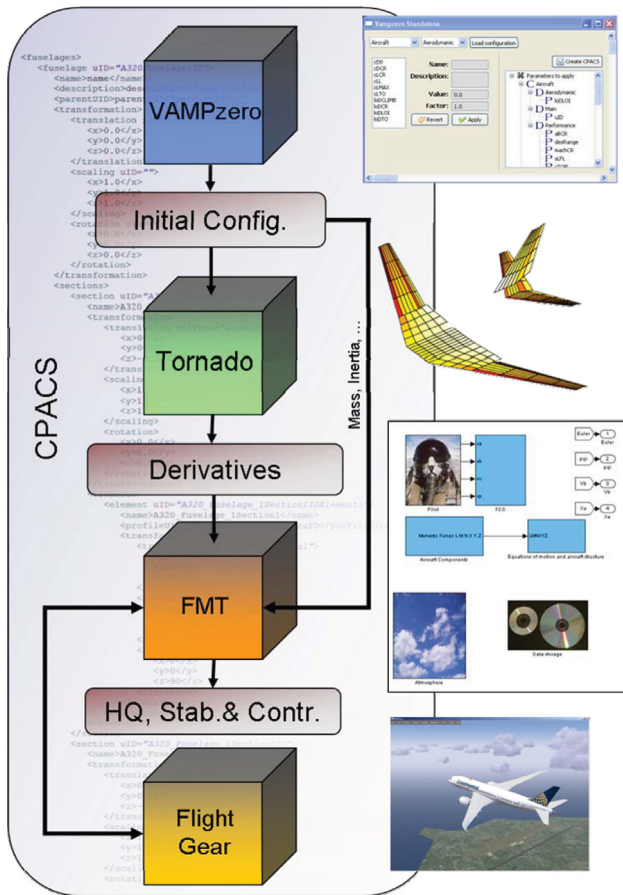

FIGURE 5. Example Framework with Tornado as Aerodynamic Method

4.1. Conceptual Design Tool "VAMPzero"

The overall aircraft data required for the here presented analysis chain can be loaded from a database or can be generated by an aircraft design tool. As 'instantiator' for novel aircraft configurations, the conceptual aircraft design tool VAMPzero was integrated.

VAMPzero was developed in course of the DLR internal project VAMP (Virtual Aircraft Multidisciplinary Analysis and Design Processes) for applications in CPACS-based environments. Input and output are based on CPACS conventions hence it can directly be integrated into the analysis chain without wrapping.

The code comprises design modules based on empirical and elementary physical correlations as they are state-of-the-art in conceptual design. The object oriented realization permits to use the design modules in a component or discipline driven way and to automatically

adapt the structure of the calculation process to the type of input specified. VAMPzero flexibly handles any combination of mission performances and design parameters as input and calculates all data not specified as long as the problem is neither under constrained nor over constrained.

Aimed at variable fidelity applications, VAMPzero can be used for the synthesis of higher fidelity MDO chains which do not cover all aspects of overall aircraft deign. Internal methods can be substituted by external measures in the calculation flow making use of knowledge based approaches to adapt conceptual and, to some extent, higher fidelity methods. Besides design variables and performance data, results comprise sensitivities and a mind map based data model which permits to retrace the actual calculation history. [5] [6]

As native component of CPACS based frameworks, VAMPzero is directly integrated into the here presented analysis chain. It is used to generate configurations for typical Top Level Aircraft Requirements (TLAR). The output includes, amongst others, geometry (fuselage, wings, engine position), mass (quantities, locations and resulting inertia) on group level (structure, power units, Systems, etc.), and some basic engine information. Some definitions not yet provided by the current VAMPzero output such as control surface definitions were complemented by external knowledge patterns.

4.2. Aerodynamics (TORNADO)

Besides masses, moments of inertia and engine thrust, aerodynamic polars and derivatives are required for the flight mechanical analysis. Aerodynamic data is calculated for different flight conditions and stored in tables which are later used in the flight mechanical simulation. Derivatives are calculated for the whole aircraft but also for each wing and control surface.

General inputs for the aerodynamic methods are provided in the CPACS file comprising the geometric definition, the tool settings (tool specifics) and information of the flight conditions to be analyzed (Mach number, angle of attack, etc.). The interpretation of the defined geometry for the aerodynamic method used is dependent on the founding theoretic formulation.

In the basis analysis framework, the Vortex Lattice Method (VLM) TORNADO is implemented. This MATLAB code was developed in cooperation of KTH Stockholm, University of Bristol and Redhammer Consulting Ldt. intended for linear aerodynamic wing design applications, in conceptual aircraft design and aeronautical education. An extensive discussion of TORNADO and its integration via a specific wrapper is given in Section 5 where the integration of different aerodynamic measures is presented.

4.3. Flight Simulation (Flight Mechanics Toolbox)

The flight mechanics toolbox (FMT) enables to integrate sub models from various disciplines (Aerodynamics, Structures, Flight control, Propulsion, etc.) in a single full

nonlinear aircraft model. The aircraft model can be used to analyse the performance of the complete system. The general aim is to apply this toolbox in a design framework. Models can be created and analysed in an automated fashion which makes it possible to integrate the flight mechanics toolbox in a multi-disciplinary design optimization framework. The central part of the toolbox are the equations of motion. These are modelled by using multi-body dynamics (Matlab SimMechanics). This enables the user to include e.g. airframe flexibility during manoeuvres, morphing aerostructures and landing gear models. The toolbox is set up in a modular structure which makes it relatively straightforward to improve the fidelity of the models and to extend the capabilities. Finally, it should be noted that the toolbox can be used for both aircraft and rotorcraft analysis. Some applications for which the FMT has been used over the past years are described in [14], [15] and [16].

The FMT input is dependent on the type of analysis which should be done. For the analysis of the trimmed condition, time domain simulations and handling qualities the input includes derivatives of forces and moments and delta values of control surfaces in several flight conditions, reference values (area, length), inertia matrix, aircraft mass and control surface information (number, task, constrains).

The FMT Wrapper reads the xml CPACS file with the xml2mat Matlab script. The output of the script is a Matlab data structure (struct) which includes all data of the CPACS file. From this struct, each parameter can be read and used in Matlab. The input format of the FMT is a known struct and is setup by a Matlab script using the CPACS parameters. Each parameter which is not in the CPACS file is hardcoded.

Tool outputs are written directly into the CPACS struct. With the mat2xml Matlab script the whole CPACS XML file is write again including the new information.

4.4. Flight Visualization (FlightGear)

Flight Gear is an open source flight simulator and has the capability to be used as visualization tool for pre-calculated flight motions. A Matlab/Simulink Toolbox is available for direct interfacing with flight gear. With a direct link to the FM-ToolBox, the motions can be visualized in real time. For piloted simulations, aircraft control hardware such as joysticks or even a flight simulator can be used with the Simulink model.

4.5. Realization and Exemplary Results

A framework for flight mechanical analysis in the preliminary stages of aircraft design was realized using the CPACS format as backbone and MATLAB as numerical platform.

The modules are implemented with individual wrappers but with a similar structure. For the programming of a wrapper it is necessary to understand the parameterization of the modules, further to understand what is the input and what is the output, how the input can find and obtain information from the exchange data format and how the

output can write back to the exchange data format. The effort decreases considerably with experience because each wrapper has a similar structure and similar parameter concepts of tools.

The flight mechanics framework was used to prove the functionality. A baseline test configuration similar to an A320 aircraft configuration and a modified version with a different sweep angle than the test configuration was chosen. The baseline configuration has a leading edge sweep angle of 27° and the modified one of 0°. Both configurations are shown in FIGURE 6. The reference point and the inertias are in both configurations equal.
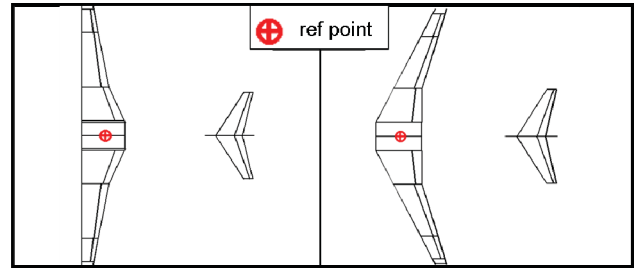

FIGURE 6. Example configuration with 0° (left) and 27° (right) sweep

The aerodynamic analysis was done in Tornado at a flight condition of 11000 meters and airspeed of 200 m/s. An analysis of the FM-tool is shown in FIGURE 7. As example, a time domain simulation of a longitudinal stick step input was performed. In the first second of the simulation, the controls are kept in the trim position. The modified configuration shows oscillations which is an effect of the forward moved neutral point, further a reduction of the sprig stiffness of longitudinal motion and so a reduction of the static length stability.
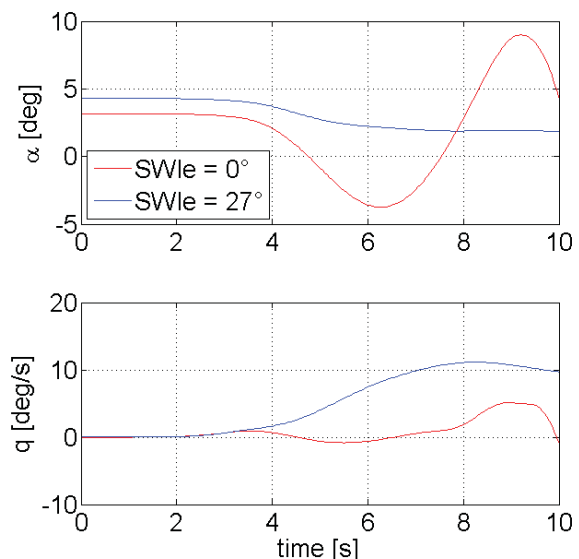

FIGURE 7. Time domain simulation of example configuration with 27° and 0° sweep; AS = 200 m/s, Altitude = 11000 m

The calculation time of the framework is around sixe minutes. The most time consuming part with around fife minutes, is the Tornado calculation which is again dependent of the numbers of used panels and the number of flight conditions. The calculation time of VAMPzero is around 10 seconds and of the FM-ToolBox 60 seconds.

5. INTERCHANGEABLE AERODYNAMIC FRAMEWORK COMPONENTS

The final result of this analysis chain, which is the flight mechanical evaluation, is directly dependent on aerodynamic result data. Thus, high fidelity aerodynamic measures are aspired with regard to the quality of results or if unconventional configurations are of concern while in the same time requirements on low computation time lead to fast and subsequently simple measures. Interchangeable aerodynamic framework components would permit to choose the method according to the character of the application - if the impact of uncertainties and error propagation on the accuracy of results is determined. However, different aerodynamic measures imply different types of input data required and output data which can be returned to the framework. Therefore variable fidelity approaches constitute an own challenge from the integration point of view.

In this section, the integration of variable fidelity aerodynamic tools is investigated. Each tool requires a specific wrapper component which interfaces the global aircraft parameterization with the tool specific variable space.

The CPACS definition includes the information of a full outer surface of an aircraft which can be used for aerodynamic methods. Dependent of the method, the outer surface can be used with limited effort or it may have to be simplified. Higher fidelity methods tyically use the full geometry definition of the outer surface. Lower fidelity methods usually handle with section parameterization which has to be deriven from the outer surface.
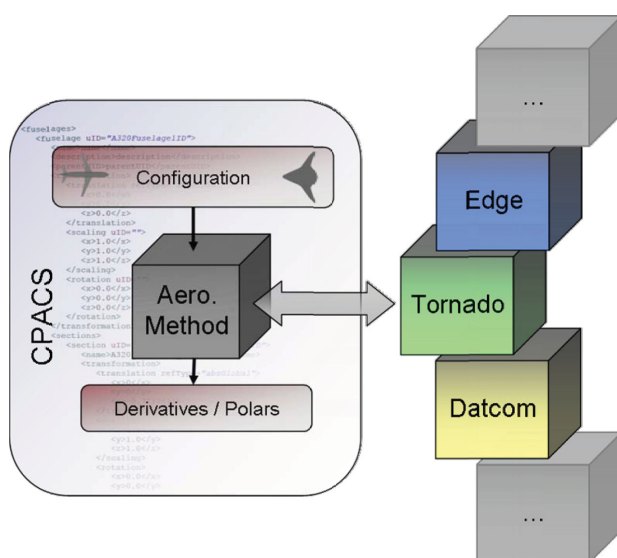

FIGURE 8. Current Aerodynamic Tool Portfolio

The wrapped components have similar input and output and mainly differ in computation time and reliability. From the integration point of view, they are interchangeable and the aerodynamic component in the analysis chain in Section 4 has to be considered as prototype for an aerodynamic method which can be selected from a portfolio (FIGURE 8).

The aerodynamic tool portfolio considered in this study is a mix of methods with different fidelities which includes DATCOM as semi-empirical, Tornado as low-fidelity and EDGE as a higher-fidelity aerodynamic method.

5.1. Semi-empirical aerodynamics (Digital DATCOM)

Digital DATCOM are semi-empirical methods for the prediction of aerodynamic stability and control characteristics, static stability, high-lift and control device effectiveness and dynamic-derivative characteristics. Mainly based on empirical data, the application of the methods is restrained to traditional aircraft concepts. The Inputs are geometrical descriptions and the outputs are the stability derivatives. Digital DATCOM can analyze wing, horizontal tail, vertical tail, body components. The analysis of control surfaces is also possible. [3]

Digital DATCOM is a FORTRAN coded executable. The input and output files are in text format which can be read and written by with Matlab for automatic DATCOM executions.

The DATCOM wrapper reads the CPACS file as described in Section 4.3. From the CPACS geometry information a Matlab script generates a DATCOM input file (CAD.dcm and for005.dat). The definition of the DATCOM input file is described in [3]. Accounting for the coarse geometry description used by DATCOM, the input file makes use of a knowledge pattern to create an equivalent input file for the potentially more complex aircraft description contained in the CPACS file. One limitation of DATCOM is for example the maximum segmentation of the wing which is three segments (wing root, wing tip and a wing king between). So the wing definition is broken down in a tree section by using mean values of geometry parameters.

The DATCOM output is also a text file (for006.dat) which is returned using a Matlab script, too. Internally data are transferred to structs and are then written in CPACS for utilisation by other tools.

Methods for reading and writing the DATCOM in- and output files are similar to the methods which are used in CEASIOM.

5.2. Low fidelity aerodynamics (Tornado)

Tornado is an open source software program, first developed in 2001 by T. Melin at the Royal Institute of Technology (KTH) in Stockholm, Sweden and is still developed further. It is based on the Vortex Lattice Method and coded in MATLAB, for linear aerodynamic wing calculations and developed for conceptual AC design. It computes aerodynamic coefficients, stability derivatives, zero lift drag (profile drag) and has a compressibility effect

correction (2D-Prandtl-Glauert). The geometric factors that can be manipulated are the wing sweep, taper ratio, dihedral, twist, different airfoils (camber) and the integration of trailing edge control surfaces. The wings are composed of trapezoids which are linked together to create a wing with different sections. The resulting quality of Tornado is comparable to other commercial VLM programs and agrees well with existing experimental data. The limitations of Tornado are, like all other VLM codes, the calculation of flow separation and transonic flight regime hence Mach numbers higher than ~0.6. [1] [2]

The Tornado wrapper reads the CPACS file as it was described in Section 4.3. A Matlab script reads the geometry information from the CPACS struct and calculates the section parameters. Tornado use as input two structs which include the geometry definition (geo.mat) and the flight conditions (state.mat). The geometry struct includes only names of airfoils which are read during the analysis form a tornado airfoil library. The airfoils which are defined in the CPACS file are added to this library automatically by the wrapper.

The Tornado input definition allows a high modelling flexibility. There is no constraint on the number of wings or on number of wing sections. Also unconventional configurations like Box Wing or Blended Wing Body (BWB) can be analysed within the capabilities of VLM. However, the input conventions imply some modelling requirements to the wrapper like profile definitions solely parallel to the X-Z plane. Further more, control surfaces are constraint to be trailing edge devices and only one generic flap type is available. The fuselage can be modelled in an elementary way and the spanwise and chordwise panel size can be controlled via tool specific data in the CPACS format.

The Tornado output is also a struct (results.mat) which is then included into the CPACS struct and finally written back to the CPACS XML file.

The principle architecture of the described Tornado wrapper is similar to the wrapper used in CEASIOM. The main difference is a more complex mapping which is required by the more hierarchical structure of CPACS.

5.3. High fidelity aerodynamics (EDGE and SUMO)

EDGE is a product of Swedish Defence Research Agency (FOI) and is a parallelized CFD flow solver for 2D/3D viscous/inviscid compressible flow. EDGE solves unstructured grids of arbitrary elements which allows EDGE to be used for problems of arbitrarily complex geometry. In course of the CEASIOM project a Matlab interface was created which allows preparing and running EDGE calculations [10] [11]. In this study EDGE is used as inviscid Euler solver which relaxes requirements to grid quality and computational effort.

For the EDGE grid generation, SUMO is used. SUMO is a surface modelling and mesh generation program developed by Larosterna [12]. It is coded in C++ as a graphical tool aimed at rapid creation of aircraft geometries. In SUMO, high quality aircraft geometry models are created form a set of closed spline surfaces with local smoothings. SUMO includes an automatic

unstructured surface and volume mesh generation. In this study the external mesh generator TetGen was used to create the volume mesh. [9] [12] [13]

SUMO saves the geometries as .smx file which is also based on XML technologies. This format is human readable and untreatable so the setup of the wrapper is straight forward.

The SUMO wrapper is also programmed in Matlab and starts with reading CPACS and writes the CPACS struct. The CPACS struct is then translated in the SUMO geometry definition. With the SUMO geometry definition the .smx input file is finally generated. In SUMO, the geometry is organized in the component types wings, bodies, controls and engines. The SUMO format allows a flexible implementation of the CPACS wing description (if the y-z-positioning points of the sections are continuous functions). Each section and airfoil can be rotated around three axes and be translated. The implementation of the fuselage is constraint by x-z-area symmetric fuselage frames. Control surfaces are regions which are marked as boundary condition. Engines are not implemented in the wrapper yet. The transition between each section is no specific regulation in CPACS. In SUMO there can chose between linear, cubic b-spline surfaces, and bicubic spline. SUMO can be run in batch mode and after the generation of the mesh files, they will copy to the working directory of EDGE. SUMO contains a direct interface for the CEASIOM data format.

EDGE can be executed from a MATLAB interface. An input file with analysis setting and flight conditions is automatically generated by the wrapper. The output of EDGE is saved in the project folder. The output is, among others, a struct with aerodynamic data and a Tecplot file. The aerodynamic data is added to the CPACS struct and translated back to the xml format.

5.4. Realization and Exemplary Results

This section shows the exemplary application of the wrappers. As test case a conventional narrow body configuration similar to the Airbus A320 was used. The basis configuration was created using the conceptual aircraft design tool VAMPzero. The following results are presented exclusively to illustrate the successful implementation of the different fidelity wrappers. The discussion of the physical differences between the different codes and dedicated validation will be presented in a separate publication.

5.4.1. Comparison of geometries

One efficient approach to find bugs in the translation between the different parameterizations is by comparison of the resulting geometries. The reference geometry in CPACS was plotted using the dedicated TILG viewer [21]. The reference aircraft is shown in top of FIGURE 9. The second plot shows the SUMO created geometry which equals the TIGL geometry. The plot below shows the Tornado geometry with the equal planform of wings and empennage but with the zero thickness representation implied by VLM. At the bottom the DATCOM geometry illustrates the simple representation as classically used in conceptual aircraft design eg. wings being considered to

be single trapezoids. Thus, differences in geometry result from different physical models used. Comparison of quantities which should match between the models are in good agreement implying a correct implementation of the wrappers.
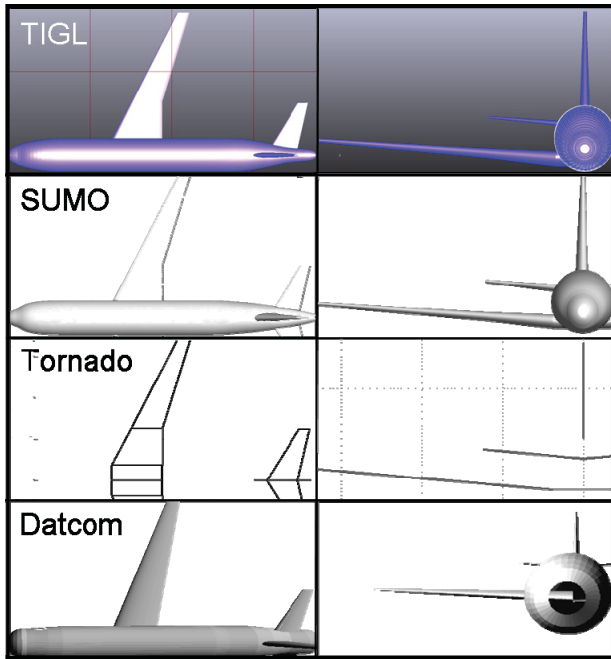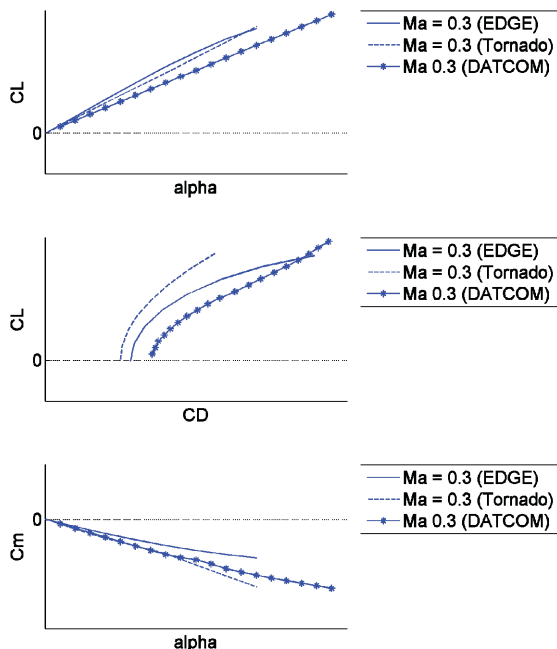

FIGURE 9. Comparison geometries

FIGURE 10.   Comparison of DATCOM, Tornado and EDGE at Ma = 0.3, Altitude = 11000 m

5.4.2.   Aerodynamic comparison

An exemplary comparison of all three aerodynamic methods results is shown in FIGURE 10. The flight condition was at a flight altitude of 11000 meter and a Mach number of 0.3. The reference area, length and point was always the same. All three methods showed an agreement at low alphas but always with different C_{D0} values as it was expected for the different theories employed. The DATCOM drag estimation included friction drag estimation. Tornado and EDGE include no friction drag estimation. The resulting reasonable differences in the results imply a correct integration of the analysis codes. The comparison of the physical results and approaches how to combine the methods in the best way will be subject of a later publication.

6.   CONCLUSION AND OUTLOOK

Conclusion

This paper presented the implementation of a framework with modules for conceptual aircraft design, aerodynamic analysis, flight mechanical analysis and visualization. The focus was placed on the technique for the integration of the different methods which were sourced from different frameworks hence contained heterogeneous interfaces.

The backbone of this study was the Common Parametric Aircraft Configuration Schema (CPACS) which served as central data model. Modules from different aircraft design frameworks (DEE and CEASIOM) and of different fidelity level (DATCOM, VLM, EULER) were integrated into the CPACS platform. The comparison of the first results implies that the creation of the wrappers and the integration was realized correctly.

The procedure to set up a wrapper always follows a standardized routine: identify the in- and outputs of the tool, select the needed data from the exchange file and converge them in the right tool input format, converge the tool output file back to exchange file format. The most effort was always in finding and understanding the relevant tool input parameters. The best support can always be found in the tool handbook (if there exist one) or in example input files. The experience from CEASIOM was also useful.

The best integration strategy depends on the level of fidelity employed by the different design systems. If the levels are similar, a general wrapper can be created to build a general bridge between the two systems. In the other case the transformation between the different levels of fidelity can better be handled if tools are individually wrapped.

Basically the tool parameterizations of the investigated frameworks are similar and it appears to be possible to agree on one unified standard for data exchange in preliminary aircraft design.

Outlook

The integration studies showed that the different parameterizations can widely be translated into another. Thus, an initiative was launched by DLR, TU Delft, KTH Stockholm and Stanford University to investigate the possibility to agree on one standardized data format for aircraft pre design. This would enable re-usability of tools also beyond the frame of individual projects or groups. CPACS is being developed with this intention and will be put to a test in course of this initiative.

The here presented wrapped tools and the built tool chain will be validated and will then be used for follow on research and design studies on conventional and unconventional configurations. The here developed capabilities are the basis for the investigation of the propagation of errors and uncertainties. In the next steps requirements to the accuracy of tools will be derived in order to meet certain accuracies of the overall chains. Together with the investigation of uncertainties of disciplinary capability modules, also trades between computational effort and results quality as well as reasonable combination of tools can be investigated.

REFERENCES

[1] www.redhammer.se (07.July.2011)

[2] T. Melin, A Vortex Lattice MATLAB Implementation for Linear Aerodynamic Wing Applications, Master Thesis, Royal Institute of Technology, 2000Linear Aerodynamic Wing Applications

[3] McDonnell Douglas Astronautics Company, THE USAF STABILITY AND CONTROL DATCOM Volume I, Users Manual, 1999

[4] D. Böhnke, B. Nagel, V. Gollnick, An Approach to Multi-Fidelity in Conceptual Aircraft Design in Distributed Design Environments, IEEE Aerospace Conference, 2011

[5] D. Böhnke, J. Jepsen, B. Nagel, V. Gollnick, C. Liersch, An Integrated Method for Determination of the Oswald Factor in a Multi-Fidelity Design Environment, CEAS European Air and Space Conference, 2011

[6] T. Zill, D. Böhnke, B. Nagel, V. Gollnick, Preliminary Aircraft Design in a Collaborative Multidisciplinary Design Environment, AIAA Aviation Technology, Integration and Operations Conference, 2011Arthur Rizzi, Modeling and Simulating Aircraft Stability & Control the SimSAC Project, AIAA Guidance, Navigation, and Control Conference, 2010

[7] http://www.ceasiom.com (07.July.2011)

[8] A. Puelles et al, CEASIOM XML fiel Definition, Royal Institute of Technology (KTH), Stockholm, 2010

[9] A. Rizzi, J. Oppelstrup, M. Zhang, M. Tomac, Coupling Parametric Aircraft Lofting to CFD & CSM Grid Generation for Conceptual Design, AIAA Aerospace Sciences Meeting, 2011

[10] http://www.foi.se/FOI/Templates/ProjectPage___4690.aspx (07.July.2011)

[11] FOI Defence and Security, Systems and Technology Edge Theoretical Formulation, 2007

[12] http://larosterna.com/sumo.html (08.July.2011)

[13] http://tetgen.berlios.de/ (08.July.2011)

[14] van Ginneken, D. A. J., Voskuijl, M., van Tooren, M. J. L., Frediani, A., "Automated Control Surface Design and Sizing for the Prandtl Plane," 6th AIAA Multidisciplinary Design Optimization Specialist Conference, Orlando, Florida, 2010.

[15] Kok, E., Voskuijl, M., van Tooren, M. J. L., "Distributed Propulsion featuring Boundary Layer Ingestion Engines for the Blended Wing Body Subsonic Transport," 6th AIAA Multidisciplinary Design Optimization Specialist Conference, Orlando, Florida, 2010.

[16] Voskuijl, M., de Klerk, J. and van Ginneken, D., "Flight mechanics modelling of the Prandtl plane for conceptual and preliminary design," Variational Analysis and Aerospace Engineering, Springer, Italy, September 2010.

[17] T. Zill, D. Böhnke and B. Nagel, Preliminary Aircraft Design in a Collaborative Multidisciplinary Design Environment, German Aerospace Center DLR, Hamburg, 2011

[18] G. La Rocca, Knowledge Based Engineering Techniques to Support Aircraft Design and Optimization, ISBN/EAN: 978-90-9026069-3, 2011

[19] Morris, A. MOB A European Distributed Multi Disciplinary Design and Optimisation Project. AIAA 2002-5444, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia , 2002.

[20] A. Bachmann, M. Kunde, M. Litz, A. Schreiber, Advances in Generalization and Decoupling of Software Parts in a Scientific Simulation Workflow System, German Aerospace Center (DLR) Simulation and Software Technology, 2010

[21] http://code.google.com/p/tigl/ (22.July.2011)

[22] La Rocca, G. and van Tooren, M. J. L. Knowledge-Based Engineering Approach to Support Aircraft Multidisciplinary Design and Optimization. Journal of Aircraft 46 (6) 2009

[23] http://www.phoenix-int.com/software/phx_modelcenter.php (03. August.2011)