Proc. of the Second Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-044-6 doi: 10.15224/ 978-1-63248-044-6-20

A Java desktop application for Aircraft Preliminary Design

[A. De Marco, F. Nicolosi, L. Attanasio, P. Della Vecchia]

***Abstract* — The paper deals with the ongoing development of ADOpT (Aircraft Design and Optimization Tool), a Java-based desktop application conceived as a fast, reliable and user friendly computational aid for aircraft designers in the conceptual and preliminary design phases.**

***Keywords* — Java, Aircraft Design, Software Engineering, Application of Computer Science**

I. Introduction

ADOpT (Aircraft Design and Optimization Tool) is a Java-based desktop application developed at the University of Naples Federico II, conceived as a fast, reliable and user friendly computational aid for aircraft designers in the conceptual and preliminary design phases. The ultimate goal of such a tool is to perform a parametric, multi-disciplinary analysis of an aircraft and then search for an optimized configuration. The search domain boundaries are usually defined by the user through a set of specifyed parameters.

The aircraft design process is well described in many books [1-7] that represent a fundamental know-how for engineers and professionals in the activity of design of airplanes. Since the end of 80's many software dealing with aircraft design (some of them based on the previous cited references) have been produced with the aim of having some design framework to be used for teaching and professional purposes in aircraft design [8-12]. Some new software [11, 12] have recently followed innovative approaches considering concepts like KBE (knowledge Based Engineering) and MDO (Multi-Disciplinary Optimization), have highlighted the necessity of an efficient graphical user interface and the importance of making the application results easily exploitable with external software. In the last decade the authors have been working on the development of a software tool for aircraft design named ADAS (Aircraft Design And Synthesis), to be used mainly for teaching purposes [13]. The software has been developed in Visual Basic making extensive use of an efficient and easy graphical user interface; also, it has been recently used in collaboration with other scientists for aircraft design application [14]. The development of ADAS provided a significant amount of experience in several fields, including software engineering, I/O best practices, GUI design and usability.

Agostino De Marco, Fabrizio Nicolosi, Lorenzo Attanasio,
Pierluigi Della Vecchia
University of Naples Federico II, Dept. Industrial Engineering, Italy

The software presented here is an extended version of ADAS entirely rewritten in Java and designed for industrial purposes. The choice of the programming language was driven by several considerations. These include the following:

- the language should be widely supported; this to avoid the case of many valid aircraft design applications and libraries that became obsolete due to the aging of the programming language used to build them;
- the language should promote the use of open source libraries, especially for I/O tasks and for complex mathematical operations;
- the language and the companion IDE should provide a widely supported GUI framework and a GUI visual builder;
- the language should support and promote modularity.

The Java programming language meets all these requirements: it is backed by Oracle and by a huge community of developers so it is continuously updated (Java 1.8 and JavaFX). Also, advanced and free IDEs (such as Eclipse or Netbeans) allow programmers to streamline and simplify the development process. In particular, the Eclipse IDE and the SWT/JFace libraries have been chosen to develop ADOpT and its GUI.

Being Java a pure object oriented programming language, it greatly encourages and simplifies modularization. Each module (package) can be programmed quite independently so that it is relatively easy to divide the work among several programmers. This is essential since the amount of classes and calculations needed to abstract, manage and analyze the entire aircraft is very large (presently the whole project counts about 56000 lines of code). For such a reason the establishment of common practices and the adherence to fundamental principle of software development (Don't Repeat Yourself, Separation of Concerns, Agile software development) are equally important.

An important design requirement of ADOpT is related to its interoperability with other engineering analysis tools. In fact, the application can be easily integrated into a comprehensive aircraft optimization cycle. This is made possible because ADOpT can be launched both in GUI and command line mode. Much care has been given to input/output and configuration files to increase the possible uses of the software.

Some of the complexities of modern aircraft design are shown in the flow chart of Fig. 1, which summarizes the conceptual application domain of the software presented here.

Proc. of the Second Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-044-6 doi: 10.15224/ 978-1-63248-044-6-20

The user/designer inputs include the external shape parameters of airplane's wings, fuselage, and tailplanes. These are passed to the analysis modules of ADOpT and a number of outputs are calculated. The design cycle is closed by applying a set of criteria that check if the design requirements are met.

Currently, the software is able to estimate the aircraft weight breakdown, the center of gravity location, the main aerodynamic parameters, and the stability derivatives. All these types of estimates can be usually performed using several interchangeable analysis methods. Extensive work has been performed during the early development stages to validate all the results returned by analysis modules of the application.

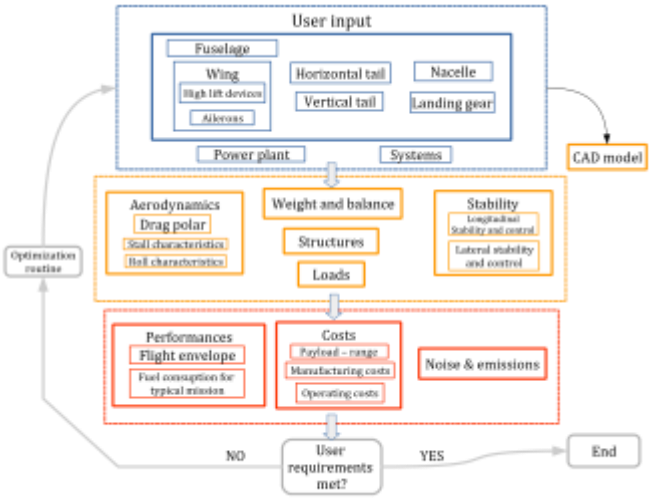

Figure 1. *ADOpT calculation modules*

II. Software architecture

The core of the ADOpT application is based on a set of three fundamental groups of Java classes. An important class is the one that manages the build-up of each single aircraft. A class hierarchy is designed to manage each aircraft subcomponents. A third fundamental class takes care of executing the required analyses (weight/balance, aerodynamic, structural, etc.) on each selected aircraft. Finally, all the objects forming an aircraft and its related analysis results are then collected and managed by a top level executive class.

A. *The top level executive class*

This class contains key methods and object for managing the application execution both in GUI and no-GUI mode:

- the main method;
- the methods needed to start the Graphical User Interface;
- a command line parser object and a command line options object in which are defined all the available command line options;
- a list containing all the aircraft created by the user;
- a list with all the operating conditions the analysis has to be performed at;
- a list with the performed analyses;
- an aircraft management object, an operating condition management object and an analysis object; these contain the current selected aircraft, the current operating conditions and the current analysis results.

The methods for starting the application in the GUI mode are defined in this class but the layout of each window is defined in the classes contained in the GUI package.

B. *The aircraft management class*

The aircraft management class is of paramount importance for the application. The main tasks of the class are:

- providing all the data of the current configuration to GUI related classes;
- providing a way to make any component aware of the existence of any other component and access its data;
- enabling the user to add or remove some components to/from the current configuration;
- providing a way to serialize all the data contained in the application.

When creating an aircraft object the user developer has to specify which components he/she wants to create (e.g., the fuselage, the wing etc.); each component is initialized with default values. The simplest way in which the default values can be overridden is by importing an XML file with custom data.

C. *The analysis management*

This is the class which is in charge of running the different types of analysis. The proper execution of an analysis requires creating an aircraft object which, together with the operating conditions, provides the data necessary to perform the computations. The class has a self-contained method for each type of analysis (e.g., weights, aerodynamics) which in turn calls all the methods necessary for a successful computation. The geometry related calculations have been put in the constructor since geometrical parameters are needed for each type of calculation; for such a reason the evaluation of any quantity should always be performed through an analysis object.

In order to obtain a reliable estimate of an aircraft MTOW (Maximum Take Off Weight), each component weight has been evaluated. An extensive literature study [1-7] has been carried out to find reliable weight estimation methods. To understand which methods were most reliable, each one's result has been compared to the component actual weight in order to evaluate relevant statistical indexes using aircraft data taken from literature. The analysis has been extremely useful to choose a set of methodologies and formulas to be used for

Proc. of the Second Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-044-6 doi: 10.15224/ 978-1-63248-044-6-20

the weight estimation of a single aircraft component. The final method for estimating a component weight is usually the average of the methods which were found to be the best with respect to the statistical indexes; it is also possible to choose a single method instead of taking the average value.

Aerodynamics surely is one of the most important drivers for aircraft design. Great care has therefore been paid to aerodynamic calculations. The authors at Department of Industrial Engineering of University of Naples have been working in activities addressed to aircraft aerodynamic design, like [15]. New methodologies to predict aircraft aerodynamics [18-22] have been recently investigated and experimentally validated through flight test activities performed on several light aircraft [16, 17]. These methodologies and have been applied in [23-25] and implemented in ADOpT.

D. *Aircraft subcomponent classes*

Since a lot of parameters have to be estimated in a single aircraft analysis, a rather nested structure has been designed to organize such computations. Each component's classes are grouped in a dedicated package. Each package typically contains:

- a main class;
- analysis classes;
- component specific classes.

Main classes hold the set of parameters that have been chosen to describe the specific component; such parameters can be easily accessed through an instance of the class wherever a valid one is available. This class is usually derived from an abstract component superclass, in which some variables that are common to each component are declared: the component's origin coordinates in the ACRF (Aircraft Construction Reference Frame), a name, a description and the component's type defined through an enumeration. The abstract component superclass has been also useful for defining the methods common to all the components. Each component main class is obviously different from the other ones; however they share some important characteristics:

- the input parameters are always stored inside the main class;
- there are no derived parameters (e.g., aerodynamics ones) except for derived geometric parameters;
- each main class provides access to all the classes in the component's package, that is, their instances are created in the component's main class and nowhere else;
- each main class does not contain inner classes.

Analysis classes hold the results of the analysis and the methods necessary to evaluate them. There is usually a class for each type of analysis, that is, aerodynamic calculations are placed in a class, weight calculations in another and so on. Many quantities (e.g., the aircraft zero-lift drag coefficient C_{D0}) can be evaluated using several different methods. For this reason all analysis classes are organized as follows:

- methods related with the evaluation of a specific quantity are usually contained inside an inner class. Each class usually contains a Map of methods which associates the method name to the value provided by the use of that method and a member function for each method; each function has to populate the Map with the proper key-value pair. Finally, a member function allows the user developer to call all the other member functions so that the results of different methods can be compared. Eventually, for those methods which have a complex structure, a dedicated inner class has been created to hold them;
- an instance of each quantity related class is initialized inside the constructor of the analysis class. This way the user developer can calculate one or more quantities by simply accessing to the related object using the getter methods;
- a method allows to execute all the functions contained in the class. This way, the user developer can execute all the available methods of the desired analysis field.

III. Software usage scenarios

A. *Execution modes*

ADOpT can be launched as a command line tool or used interactively through its dedicated graphical user interface. The command line running mode is designed for use in optimization loops typical of aircraft preliminary design (see Fig. 1).

When launched in GUI mode, the software tool gives to the user an immediate feedback about the aircraft features when input parameters are changed interactively. More than one airplane can be managed simultaneously by the application, and the analysis results on all selected design configurations can be compared side by side.

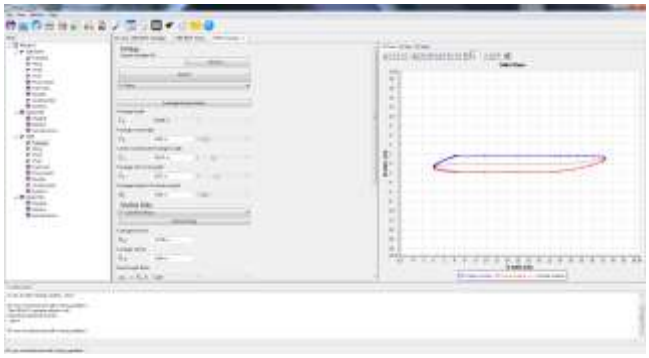

Figure 2. *Two (or more) different components (eventually belonging to different aircraft) can be opened and managed through the application frontend.*

Proc. of the Second Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-044-6 doi: 10.15224/ 978-1-63248-044-6-20

The GUI features a 3D viewer widget showing the CAD model of the aircraft, which is generated using the well known Open CASCADE Technology libraries [26]. The CAD representation can be eventually exported in various file formats for further usage in other CAE applications.

B. *Input and Output files*

Regarding the Input/Output capability of ADOpT, the application accepts configuration files in XML (eXtensible Markup Language) format and exports the results on file in two possible formats: XML and Microsoft Excel (XLS). An example of output in Excel format with results of four different analyses on four different airplanes is shown in Fig. 2.

The XML input files are also used in interactive work sessions to import a predefined aircraft and populate the GUI controls accordingly. The entire application status can be saved to a XML file generated through the XStream library [27] which provides serialization and deserialization functions; this file can then be used to restore the application status at a later time.

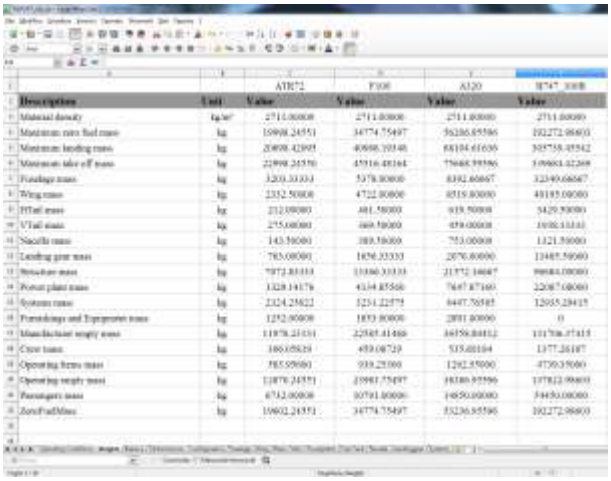

Figure 3. *The weights breakdown sheet obtained as output after a batch analysis.*

IV. Interoperability with other analysis tools

The ADOpT application can also be used as a basic parametrical CAD modeler to explore new designs.

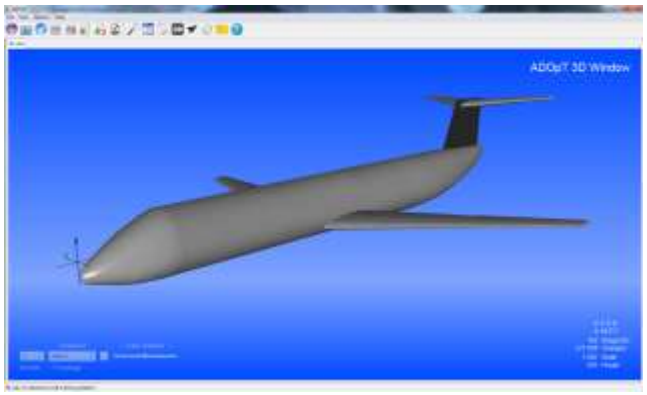

Figure 4. *A Fokker 100 CAD model generated by ADOpT, as seen from the application 3D viewer window.*

The capability to change the aircraft geometric parameters using the dedicated controls in the GUI, coupled with the 3D viewer widget, gives the user the possibility to change each component shape and dimension, view the updated CAD model and eventually export it to file once some satisfactory results have been obtained. The CAD model can be then used to further study the aircraft in a CAE suite (see Fig. 4).

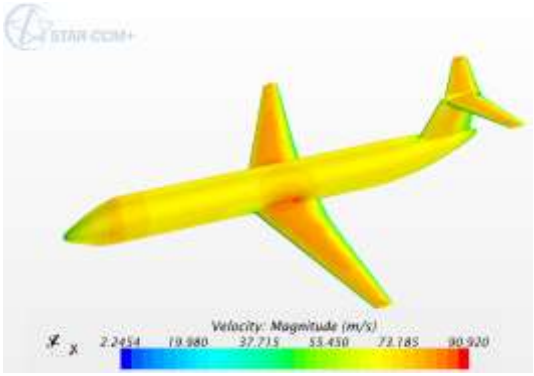

Figure 5. *The Velocity magnitude over an ADOpT-generated Fokker 100 CAD model obtained using the STAR-CCM+ suite.*

Most of the existing aircraft design software cannot be easily integrated in a comprehensive development cycle due to the lack of standard input/output files and of a command line mode. For such a reason the ADOpT application provides several possibilities:

- to read from and write to a standard format named CPACS (Common Parametric Aircraft Configuration Schema); this interchange format comes from a recent European effort for the standardization of aircraft abstraction model [28];
- to write a Datcom [29] input file;
- to write a FlightGear/JSBSim [30] input file.

These capabilities will likely extend both the possible usages of the application and its lifespan.

Proc. of the Second Intl. Conf. on Advances in Information Processing and Communication Technology - IPCT 2015

ISBN: 978-1-63248-044-6 doi: 10.15224/ 978-1-63248-044-6-20

v. Conclusions

An aircraft design and optimization desktop application written in Java has been introduced. The adoption of established software engineering practices, the use of advanced development tools, and concurrent development enabled the developer team to build a feature-rich application in a relatively short period of time. As of its design, the application is easily maintainable and extendable. The cautious choice of software dependencies and the number of features implemented will likely extend both the possible usages of the program and its lifespan.

References

[1] Roskam, J., Airplane Design Part I-VI, DAR Corporation, Lawrence, KS, 2000.

[2] Torenbeek, E., Advanced Aircraft Design, Wiley, Delft, The Netherlands, 2013.

[3] Nicolai, L. M., Charichner, G. E., Fundamentals of Aircraft and Airship Design, AIAA Education Series, AIAA, Reston, Virginia, 2010.

[4] Raymer, D. P., Aircraft Design: a Conceptual Approach, 2nd ed., AIAA Education Series, AIAA, Washington, DC, 1992.

[5] Kroo, I, Aircraft Design: Syntesis and Analysis, digital textbook, http://adg.stanford.edu/aa241/AircraftDesign.html

[6] Jenkinson, L. R., Simpkin, P., Rhodes, D., Civil Jet Aircraft Design, Arnold Publishers, London, Great Britain, 1999.

[7] Sforza, P., Commercial Airplane Design Principles, Elsevier, Oxford, UK, 2014.

[8] Advanced Aircraft Analysis. (AAA) Version 3.4, DARcorporation, Lawrence, Kansas 66049, http://www.darcorp.com/Software/AAA/

[9] Raymer D P., RDS-STUDENT: Software for aircraft design, sizing, and performance. American Institute of Aeronautics and Astronautics Inc., Reston, 1999.

[10] Scholz D., "PreSTo The Aircraft Preliminary Sizing Tool," 10th European Workshop on Aircraft Design Education (EWADE), Dipartimento di Ingegneria Aerospaziale – University of Naples, Naples, May 2011

[11] Anemaat, W. A. J., Kaushik, B., Hale, R.D., Ramabadran, N.,"AAARaven: Knowledge-based aircraft conceptual and preliminary design," 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Waikiki, HI, United States, 23-26 April 2007, Volume 7, 2007, Pages 7143-7167

[12] Piano Software for Windows, Version 5, Lissys Ltd, Paterson Drive, Woodhouse Eaves, LE12 8RL, United Kingdom, http://www.lissys.demon.co.uk/Piano5.html

[13] Nicolosi F., Paduano G., "Development of a Software for Aircraft Preliminary Design and Analysis," 3rd CEAS Air and Space Conference, Venice (Italy), 2011

[14] Zhang, M., Rizzi, A. W., Nicolosi, F., and De Marco, A., "Collaborative aircraft design methodology using ADAS linked to CEASIOM," Proceedings of the 32nd AIAA Applied Aerodynamic Conference, AIAA, Washington, DC, 2014.

[15] Coiro, D.P., Nicolosi, F. Grasso, F. "Design and Testing of Multi-Element Airfoil for Short-Takeoff-and-Landing Ultralight Aircraft, " Journal of Aircraft, Vol. 46, N. 5, Sept.-Oct. 2009, pp. 1795-1807. ISSN 0021-8669. doi: 10.2514/1.43429.

[16] Nicolosi, F., De Marco, A., Della Vecchia, P. "Flight Tests, Performances and Flight Certifcation of a Twin-Engine Light Aircraft, " Journal of Aircraft, Vol. 48, N. 1, January-February 2011, pp. 177-192. ISSN 0021-8669. doi: 10.2514/1.C031056

[17] Nicolosi, F., De Marco, A., Della Vecchia, P. "Stability, Flying Qualities and Longitudinal Parameter Estimation of a Twin-Engine CS23 Certified Light Aircraft, " Aerospace Science and Technology (Elsevier) AESCTE 2734, Vol. 24, N. 1, January–February 2013, pp. 226-240. ISSN 1270-9638. http://dx.doi.org/10.1016/j.ast.2012.12.006

[18] Nicolosi, F., Della Vecchia, P., Ciliberti, D. "An investigation on vertical tailplane contribution to aircraft sideforce, " Aerospace Science and Technology (Elsevier) AESCTE 2873, Vol. 28, N. 1, July 2013, pp. 401–416, ISSN 1270-9638. http://dx.doi.org/10.1016/j.ast.2012.12.006

[19] Nicolosi, F., Della Vecchia, P., Ciliberti, D. "Aerodynamic interference issues in aircraft directional control, " ASCE's Journal of Aerospace Engineering, Vol. 28, N. 1, January 2015, ISSN 0893-1321, DOI 10.1061/(ASCE)AS.1943-5525.0000379, 04014048

[20] Della Vecchia, P., Nicolosi, F. "Aerodynamic guidelines in the design and optimization of new regional turboprop aircraft," Aerospace Science and Technology (Elsevier) AESCTE Vol. 38 , October 2014, pp. 88-104, ISSN1270-9638. http://dx.doi.org/10.1016/j.ast.2014.07.018

[21] Nicolosi F., Della Vecchia P., Ciliberti D., Cusati V., "Development of new Preliminary Design Methodologies for Regional Turboprop Aircraft by CFD Analysis",29th ICAS Conference, St. Petersburg (Russia), 07-12 September 2014, pp. 1-11. Proceedings ISBN 3-932182-80-4

[22] Blackwell, J.A. Jr, A Finite-Step Method for Calculation of Theoretical Load Distributions for Arbitrary Lifting-Surface Arrangements at Subsonic Speeds, Washington, D.C., 1969.

[23] Nicolosi F., Della Vecchia P., Corcione S."Aerodynamic Analysis and Design of a Twin Engine Commuter Aircraft,"28th ICAS Conference, Brisbane (Australia), 23-28 September 2012, Vol. 1 pp. 321-332. Proceedings ISBN 978-0-9565333-1-9

[24] Nicolosi, F., Della Vecchia, P., Corcione, S. "Design and Aerodynamic Analysis of a Twin-engine Commuter Aircraft, " Aerospace Science and Technology (Elsevier) AESCTE, Vol. 40, Jan. 2015, pp. 1-16, ISSN 1270-9638. http://dx.doi.org/10.1016/j.ast.2014.10.008

[25] Nicolosi F., Corcione S., Della Vecchia P.,"Commuter Aircraft Aerodynamic Design: Wind-Tunnel tests and CFD Analysis", 29th ICAS Conference, St. Petersburg (Russia), 07-12 September 2014, pp. 1-13. Proceedings ISBN 3-932182-80-4

[26] Various Authors, Open CASCADE Technology, 3D modeling & numerical simulation, http://www.opencascade.org/

[27] Walnes, J. et al., XStream, http://xstream.codehaus.org/

[28] B. Nagel, D. Böhnke, V. Gollnick, P. Schmollgruber, A. Rizzi, G. La Rocca, J.J. Alonso, Communication in Aircraft Design: "Can we establish a Common Language?," 28th International Congress of the Aeronautical Sciences, Brisbane, Australia, 2012.

[29] Hoak, D. E., et al., "The USAF Stability and Control DATCOM," Air Force Wright Aeronautical Laboratories, TR-83-3048, Oct. 1960 (Revised 1978).

[30] Berndt, J. S., De Marco A., "Progress on and Usage of the Open Source Flight Dynamics Model Software Library, JSBSim." AIAA-2009-5600. Proceedings of AIAA Modeling and Simulation Technologies Conference and Exhibit 10-13 August 2009, Chicago, Illinois, USA.