



Integrated Aircraft Design Network

Raghu Chaitanya. M. V, Ingo Staack, Petter Krus

Linköping University, SE-58183, Linköping, Sweden

raghu.chaitanya@liu.se

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

Abstract

This paper describes the XML based multidisciplinary tool integration in a conceptual aircraft design framework, developed by the Division of Fluid and Mechatronic Systems (FluMeS), Linköping University. Based on a parametric data definition in XML, this approach allows for a full 3D CAD integration. The one-database approach, also conducted by many research organizations, enables the flexible and efficient integration of the different multidisciplinary processes during the whole conceptual design phase. This central database approach with a detailed explanation of the developed geometry description and the data processing, focusing on the CAD integration is presented. Application examples of the framework are presented showing the data build up and data handling.

1 Introduction

Information is generated by tools, normally coupled towards intern or proprietary data structures. In a multidisciplinary design process information has to be propagated among tools and has to be fully accessible to any tool at any time. This dilemma leads either to a central “one-tool” or a “one-database” approach.

The one-tool approach cannot solve the problem since it is not a practical and justifiable

solution as different applications need different tools for different needs. Hence the best and optimal solution would be the one-database approach. RAPID and Tango are tools are being developed in CATIA and Matlab respectively, to address one-database approach. In order to maintain flexibility and allow the developer to choose the preferred work method, both programs are implemented in parallel. Switching between the two is possible at any time [1]. Within this framework traditional handbook methods [2] till [4] are employed in the design. Similar aircraft conceptual design programs, [5] till [9] are developed by research institutions, universities and companies.

2 Data Management

Data created in RAPID or Tango can be delivered in the “eXtensible Markup Language”, (XML) format [15]. XML allows applications to represent electronic documents or text data in an easy to understand and transferable format between programs.

XML is made up of markup tags and data to represent the information. An XML forms a tree structure, this makes it easy to retrieve data and find relationship between different information represented in the XML.

Transformation of XML Document is favorable performed by XSL Transformations (XSLT). XSLT uses XPath language to navigate in XML documents. It can serve for complex

translations such as element and attribute editing (add, remove, replace), rearrangement, sorting, perform tests and make decisions [15].

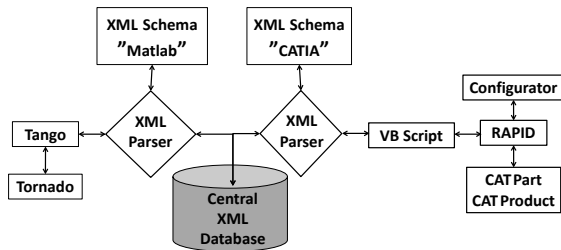


Fig. 1 XML data flow the two main applications RAPID and TANGO with the help of XSLT.

2.1 XML Integration

2.1.1 RAPID XML Export

Excel Visual Basic Application (VBA) is used to configure the CATIA [16] parameter or geometrical sets and generates into an XML. The following steps are implemented in creating XML from RAPID

- **Configuration of Parameter and Geometric sets through Excel:**

Configuring the parameters through Excel will reduce the effort of adding changes to the code whenever a new parameter/geometrical set need to be added to the XML. The configuration contains three main parts in Excel:

Parameter String: represents the parameter set/geometrical set from CATIA. All the parameters within the parameter set will be made as XML. Example:

```
reference\inputparameters
```

Parameter Array: used for making XML tags to the parameter sets or geometrical sets and parameters. Depending on the depth of the XML tree, number of values in the Array String is needed. Example: XML node-
`<part name="reference">`

Array List: Needed to put together parameter sets from the same part into one list. Each part will have one corresponding Array List. Example:

```
fuselage\inputParameters\&
"fuselage\instantiatedGeometry\"
should come under the same part
fuselage in XML. So they have same
array list Name fuselageList.
```

- List "Hash", is a dictionary object [18] (key, value) that is used in the code because of the Array List column that comes from Excel. The Array List column gives only Array List Names but does not create them. This handles the grouping of the parameter sets with the same Array List into one list instead of separate ones. For the first time an array list for a given name is created, thereafter it does not create a new array list if the name already exists.

- **Value Parsing:**

To parse the CATIA parameters and translate them to XML, two loops are used: The outer loop runs through all the parameter lists from CATIA and inner loop, runs through the parameter list of the Excel sheet. Strings from CATIA are compared with strings from Excel and the matching strings from CATIA are created as XML.

- **Writing into XML using DOM Object:**

For efficient XML editing, Microsoft. XMLDOM object is used in the VBA section to translate the parameter/geometric sets into XML. The DOM object creates the XML file and takes care of the formatting and structure. This data set tree related access method also helps in modifying the XML without any hassles or cumbersome coding.

- **Spline from CATIA to XML:**

The spline in CATIA cannot be handled similar to that of the rest of the parameter/geometrical sets, as the coordinates of the points in the splines are not available directly in CATIA tree. The geometrical set in CATIA is taken

as a parameter string from Excel, where the sketches are added. One array each is used to store the x, y, z coordinates of the spline control points from the exchange Curves. Example: "fuselage\exchangeCurves".

- Finally the XML DOM object is written to file and saved as XML.

2.1.2 XML to RAPID Import

The following steps are performed to read XML to RAPID:

- Parsing the XML using DOM object:**
 The XML file is loaded into a DOM Object. This DOM object is parsed for the required information and tags.
- Recursive Function to get child nodes:**
 The values of parameters that need to be updated into CATIA are stored in the child nodes with a `value` tag. A recursive function is used to get all the child nodes with a `value` tag from the XML and the corresponding text in these nodes. These texts are the new values to be updated.
- Constructing the Parameter Strings to be updated:**
 The parameter string that needs to be updated in CATIA needs to be constructed; a recursive function is used to get the parent nodes for the child nodes with the `value` tag. These parent node tags are appended along with "\" to form the string.
- Spline from XML to CATIA:**
 The existing spline is deleted first. The spline values in XML are stored as Control Points with x- and y-value Nodes (`xvalue`, `yvalue`). These Nodes are recursively read using DOM object. Using these x/y points, new Control Points are formed and then a new spline is created in CATIA

- Updating CATIA:**

The parameter Strings along with the corresponding values are updated in the CATIA using VBA – CATIA Functions.

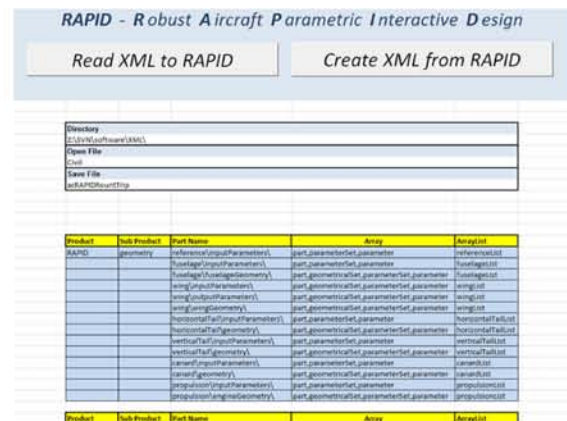


Fig. 2 The RAPID-XML Interface

2.2 Tango XML

Tango makes usage of the underlying Java DOM application classes in Matlab that serves for the XML data handling. This data is handled object oriented within geometrical or functional classes, so that every class includes the class-related XML parsing functionalities. This method allows for greater flexibility and fast replacement or appending of new classes.

The basic classes are product-geometry related arranged (e.g., wing and underlying wing partition class), whereas the higher level classes are product-functional (system) related (e.g., fuel system, primary flight control system). This class reference transition within the dataset makes it necessary to work with part pointers in order to link the functional classes with its related geometrical properties in the geometry related classes; Examples are the control surfaces (geo. def.) that are part of the PFCS (sys. def.) and the wing fuel tanks (geo. def.) that are part of the fuel/propulsion system (sys. def.). By these links, the strict hierarchal XML (tree) data structure becomes extended by cross-branch couplings, described by the part pointers.

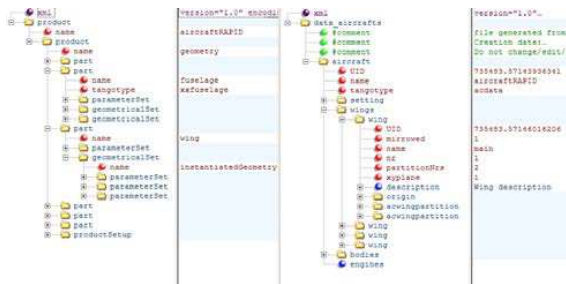


Fig. 3 Data Structure adapted towards the tools needs
(Right side: Tango XML, Left Side: RAPID XML)

3 Aircraft Geometry Data Description

Aircraft geometry is one of the most important features as it holds the entire information that is needed for the whole aircraft analysis. The aircraft data stored in XML format can be exchanged between different software, thereby decreasing the need and time to redo the aircraft.

3.1 Fuselage geometry description

The geometry is generated with the help of four splines namely upper curve, bottom curve, side curve and center curve. These splines form the base for the generation of the fuselage; later the number of frustums for the cross-sections definitions can be instantiated automatically depending on the necessity. Frustums are formed by joining two Bézier cubic curves at each end by means of a surface. The instantiated frustums can be modified to form a wide range of fuselage cross section geometries.

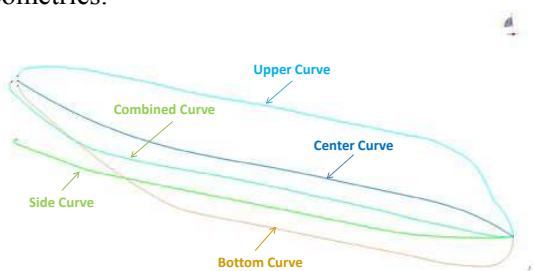


Fig. 4 The spline line fuselage curves

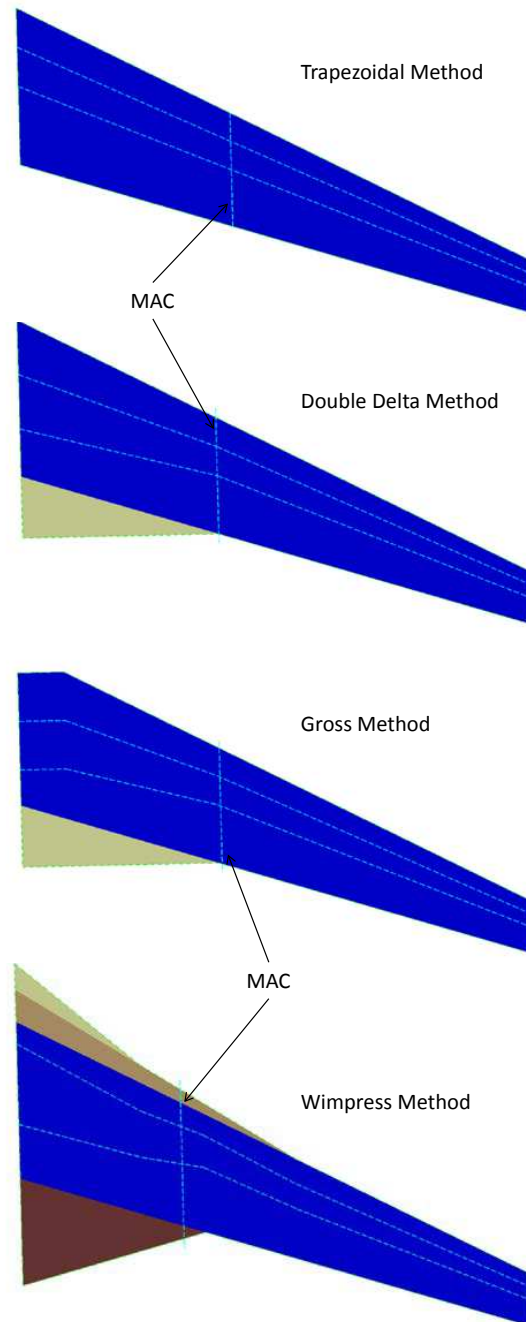


Fig. 5 The four different wing reference area methods used in RAPID

3.2 Wing geometry description

The reference area is the foundation to create a wing. Four different reference area methods are implemented [12]:

- Trapezoidal Method
- Double delta Method
- Gross Method
- Wimpres Method

During instantiation first the trapezoidal area is created, thereafter this area is used as a building block and the rest of the area methods are implemented as in Fig. 5. The wing boundary modifies itself depending on the specified reference area method. The number of wing partitions chosen by the user can be instantiated automatically. Each wing partition is formed by joining two airfoils by a surface. The airfoil is completely parametric and can be modified to obtain a wide variety of airfoils [10].

4 Framework approach

The flow of data between each discipline in a multidisciplinary design environment is coupled and saved in XML format [13] [14], and is accessible by all the required tools. The database definition (including several component libraries like functional assemblies) is parametrically defined in such a manner that a data refinement over time alongside the project is possible. In this way, a transition-less process from low or medium fidelity (in e.g. Tango) up to high fidelity (e.g., in RAPID) is realized as shown in Fig. 8

5 Application examples

This section shows the application examples of the framework, showing the data build up and data translation between RAPID and Tango and vice versa. Two examples have been tested to investigate the data flow processed in the correct approach. In RAPID, as the user has different options of reference area, this might be difficult to pick the correct method. A number of parameters are accessible for the user in order to obtain various configurations. This might lead to a geometry that is over-defined or has a lot of parameters to play with.

5.1 Example 1

In this example the double delta reference method is used. The cross-sections of the fuselage range from a circle to an ellipse. The data was successfully exchanged in both ways.

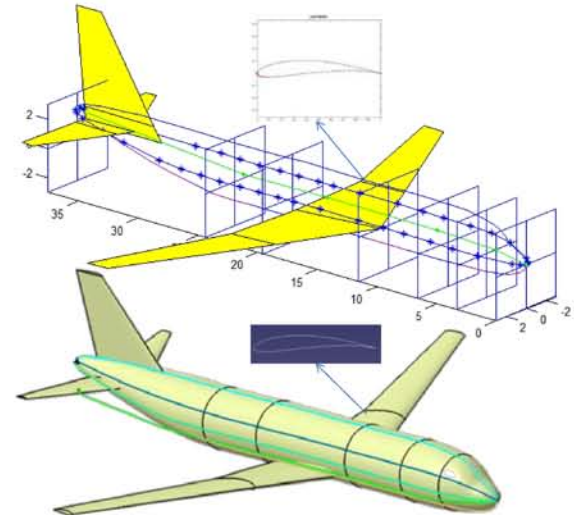


Fig. 6 Civil Aircraft in Tango (top) and RAPID (bottom)

5.2 Example 2

A much complicated fighter aircraft was selected to test as shown in Fig. 7. Data exchange showed promising results. It is to notice that the data structure in the background of both examples is similar with modified parameters with added lifting surface “canard” in the fighter example.

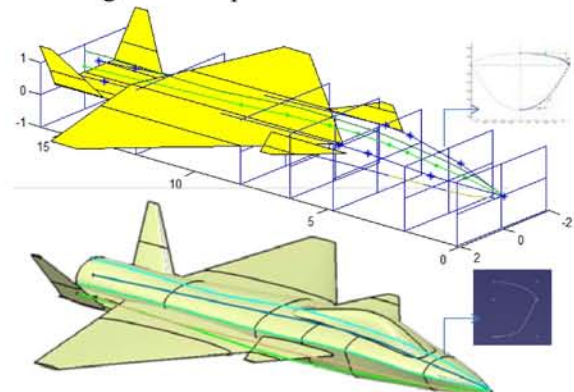


Fig. 7 Military Aircraft in Tango (top) and RAPID (bottom)

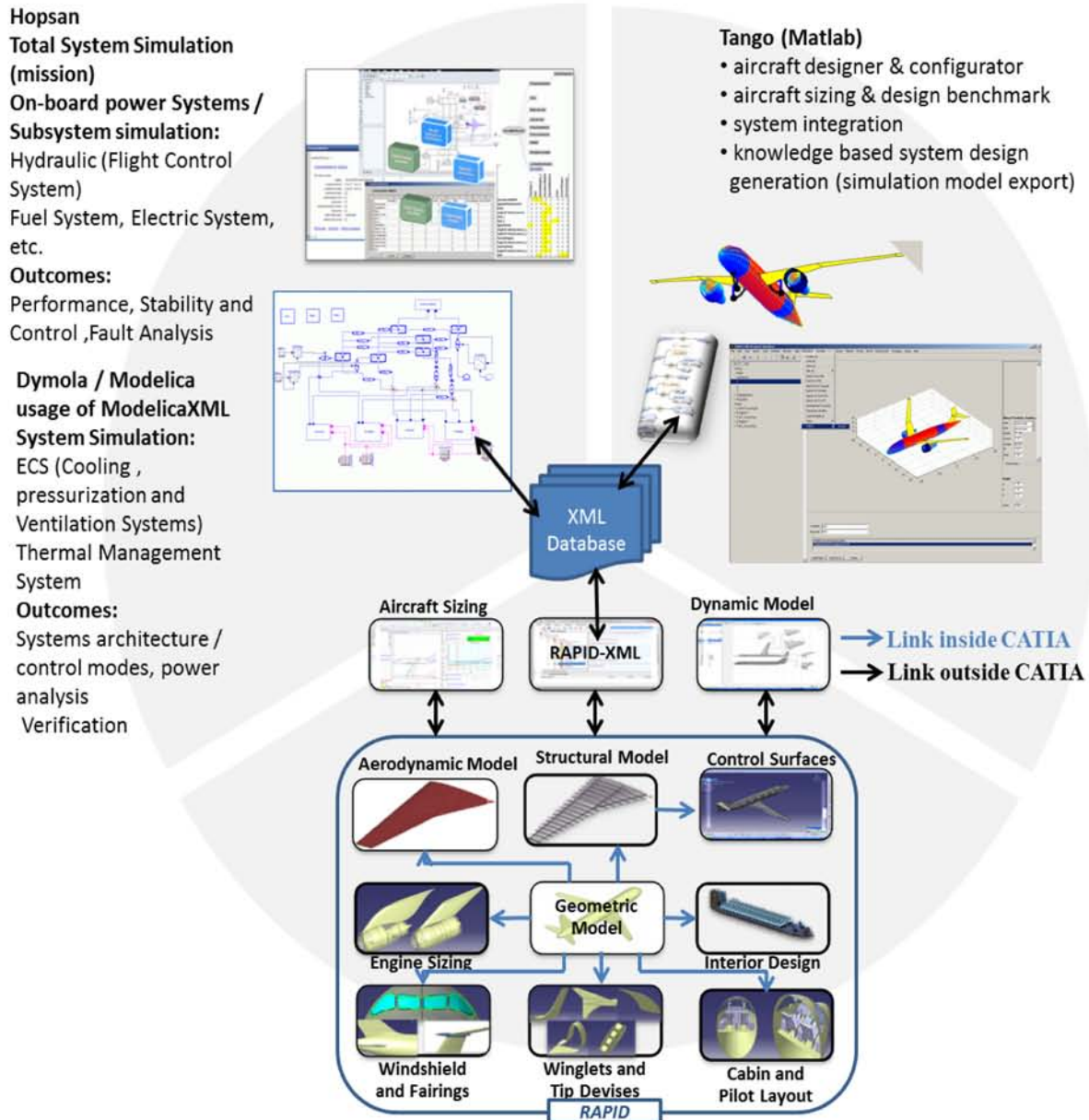


Fig. 8 Integrated Aircraft Design Network

6 Conclusion

The paper shows the multidisciplinary conceptual aircraft design analysis based on a central parametric XML database. This database -containing all project related data- is intended to grow simultaneously with the refined specification of the airplane.

Main advantage using XML is the easy and smart access from literally any programming language which makes it, together with the fact that it is human readable ASCII code, predestinated to be used in multidisciplinary and therewith multi-tool frameworks. These features serve for an easy adaptation and integration of new tools, scripts, etc. Due to the XML data tree structure, the developer has to arrange the data

in a certain position, however the strict tree structure fit not totally towards the needs of a complex product as an aircraft; here, because of the transition of the class alignment from geometrical placement (low level) towards system description (high level), these trees have to be extended by cross-nodes pointers.

The 3D CAD description of this data setup is a small fraction of the original data needed in the CATIA environment to establish the geometry; here, extensive usage of knowledge base descriptions, namely *knowledge pattern* and *power copy* concepts are used. This method limits the design space in favor of a slim dataset consisting of rather significant parameters. This allows for a direct access of the geometry for other tools, like a geometrical optimization outside the CAD environment. The unified geometry makes meshing easier and serves for no aperture for high fidelity CFD analysis.

As proposed in Fig. 6 even simulation models can be generated out of the (mainly geometry) XML aircraft description.

Acknowledgement

Funding of this work was provided by NFFP, the Swedish National Aviation Engineering Program [11]. The contributing authors wish to thank the NFFP founders for this support. The First author would like to thank Krishnaveni Chitrapu for the helpful suggestion for XML creation from CATIA.

References

- [1] Staack, I., Raghu Chaitanya, M. V., et al., "Parametric Aircraft Conceptual Design Space," *Proc 28th Congress of the International Council of the Aeronautical Sciences*, Brisbane, Australia, 2012.
- [2] Raymer D. Aircraft design - a conceptual approach, 5th edition, AIAA education series, Washington DC, U.S.A, 2012
- [3] Roskam J. Airplane design- part1: Preliminary Sizing of Airplane, DARcorporation, Lawrence, 1985
- [4] Torenbeek E. *Synthesis of subsonic airplane design*, Delft University Press, Delft, Netherlands, 1995
- [5] Raymer D. RDS-student: software for aircraft design, sizing, and performance, Volume 10, AIAA education series, Washington DC, 2006
- [6] CEASIOM. Computerized Environment for Aircraft Synthesis and Integrated Optimization Methods software, <http://www.ceasiom.com>
- [7] PADLab Software, <http://www.luftbau.tu-berlin.de/menue/forschung/padlab>
- [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] Melin, T., Parametric Airfoil Catalog, Part I, Linköping University, ISBN: 978-91-7519-656-5, 2013
- [11] VINNOVA. Swedish national aviation engineering research programme, <http://www.vinnova.se/en/Our-activities/Cooperation-Programmes/National-Aviation-Engineering-Research-Programme/>
- [12] Isikveren. A. T., Quasi-Analytical Modeling And Optimization Techniques For Transport Aircraft Design, Ph.D. Dissertation, Royal Institute of Technology (KTH), Stockholm, 2002
- [13] Risheng Lin and Abdollah A. Afjeh, An XML-Based Integrated Database Model for Multidisciplinary Aircraft Design, *Journal of Aerospace Computing, Information, and Communication* 2004 1:3, 154-172
- [14] Ho-Jun Lee, Jae-Woo Lee, Jeong-Oog Lee, Development of Web services-based Multidisciplinary Design Optimization framework, *Advances in Engineering Software*, Volume 40, Issue 3, March 2009, Pages 176-183, ISSN 0965-9978,
- [15] <http://www.w3.org/>
- [16] CATIA V5 Release 21, <http://www.3ds.com/>
- [17] Microsoft Excel, <http://www.Microsoft.com/>
- [18] <http://msdn.microsoft.com>