Automation of Aircraft Pre-design Using a Versatile Data Transfer and Storage Format in a Distributed Computing Environment

Arne Bachmann[*], Markus Kunde[*], Markus Litz[*], Andreas Schreiber[*], Lothar Bertsch[†]
[*]*German Aerospace Center*
Simulation and Software Technology
Cologne, Germany
{Arne.Bachmann, Markus.Kunde, Markus.Litz, Andreas.Schreiber}@dlr.de

[†]*German Aerospace Center*
Institute of Aerodynamics and Flow Technology
Brunswick, Germany
Lothar.Bertsch@dlr.de

***Abstract*— In the Aerospace field one often has to deal with a host of highly specialized software applications that need to be orchestrated into one optimization process to produce e. g., an optimized aircraft model. This optimization process combines engineering knowledge from fields as diverse as aerodynamics, aeroelasticity, engine building, environmental impact assessment, material science and structural issues. For each field of science there are highly performant problem solvers available, but yet they don't share the same data exchange formats. The specialized knowledge in each institution involved in a larger project cannot be leveraged easily by other project partners, thus calling for a software and data integration solution to enable global interconnection of local tools. At the German Aerospace Center (DLR) there have been carried out several projects linked with this challenge. We will present here shortly the building blocks for data interchange as well as an example for a workflow building between DLR's institutes (work in progress).**

***Keywords*—Aviation; integration framework; simulation; workflow; XML**

I. Introduction

At the German Aerospace Center (DLR) there are a lot of institutes working in different fields of aerospace science in the wide range from fundamental research to application development. In several projects described in Section II, different institutes at DLR cooperated. For this, a general data exchange and storage format was developed and is still in continous adaption to emerging requirements. This data format was deliberately given a very generic design to be independent of any implementation details of any software integration framework, firstly to cultivate a more application domain centered design and secondly to enable easy transition between, as well as adaption to any other framework to be used in the future.

While most aspects and problems connected with the very general nature of the developed data format have already been presented elsewhere [1], this paper focuses more on a real life problem solving in a custom workflow exemplified by presenting work in progress, and how our approach helps researchers in science.

The paper is organized as follows: Section II gives a deeper motivation for the work carried out and sheds some light on related projects. Section III gives a short overview of the data format and how it is used. In Section IV an example application of integrating a tool chain is shown. Section V draws a conclusion including a brief evaluation of current outcomes and expected future developments.

II. Motivation

Because interdisciplinary projects have become more common and important not only at DLR, the need for a common data exchange and problem solving platform became imminent. In each research area there are very good tools for specific problem areas available, often more elaborated and accurate than found in commercially available all-in-one solutions. The need to be able to build up inter-department cooperations quickly was well known at DLR and led to a number of projects since 2005, that were involved in developing the presented software integration development:

- *Technology integration for the virtual aircraft (TIVA/TIVA-II), 2005-2008.* To perform technology assessment regarding collaborative predesign of civil aircraft. The main goal of this project was the definition of a common parametric data model, suitable for application in the preliminary design phase. A suite of tools was adapted to this data format and integrated into a process automation framework. A follow-up project is currently being planned;
- *Climate optimized air transport system (CATS), 2008-2012.* The goal of this project is to optimize aircraft and aircraft missions regarding their climatic impact. In this project the technology described in this paper is combined with another tool developed for climate assessments;

- *Evaluation of innovative turbine engines (EVITA), 2008-2011.* The technology and data used in this project is quite similar to *TIVA*, but is much more focused on the experimental predesign of engines than on the whole aircraft;
- *Unmanned combat air vehicle (UCAV-2010), 2007-2010.* This project simulates the development of a military aircraft optimized for stealth capabilities. Existing knowledge gained in the *TIVA* project is taken into account. The central data set was adapted to the additional requirements, and an expert system was added [2].

For successful cooperation in computer-aided optimization processes and model assessments, scientists and engineers need proper data interfaces to communicate knowledge and transfer aggregated data. To do so, usually computer scientists help in designing data structures and software architecture as well as distributed computer infrastructure to create useful systems for researchers to ease their work, including computation, optimization and data management.

In their own self-contained knowledge domains all institutes and departments at DLR developed highly optimized tools, but for more global optimization tasks institutes need to cooperate. While this is basically possible using custom data converters and batch-scripts for simple tool chains – as was employed in earlier projects, including TIVA – this approach is not powerful and flexible enough for today's demands to build up collaborative projects quickly and on a firm technical ground.

III. Data integration approach

Despite the fact that in the scientific world there are already a lot of data storage and transfer formats available (cf. CDF [3], HDF [4], Step [5]), there was communion over the idea of creating a DLR-wide quasi-standard for an extensible data format, originally designed for use in the preliminary design phase of single aircraft optimization, as described in [6], but also suitable for other areas loosely connected with aviation.

A. Data transfer and storage

The result can be found in the CPACS format (Common Parametric Aircraft Configuration Schema), which is currently employed and extended in several collaborative projects at DLR. To extend the format's usefulness even further, the feasibility of a transformation between CPACS and Step formats is already in progress [7].

During earlier development stages of the tool suite the original toolchain design including a central dataset all tools were writing to as in [6] was abandoned and replaced by a streaming data transfer approach as described in more detail in [1]. The dataset is currently capable of not only holding e. g., aerodynamic and geometric data of one aircraft configuration – meaning position, shape and number of wings, engines and so on – but even data for whole fleets of aircraft (and other vehicles) as well as very detailed information about turbines – as being developed in the EVITA project described above – and cost approximations or flight mission descriptions of the CATS project. In contrast to the aforementioned data formats, the used schema builds upon human-readable XML (http://www.w3.org/XML/) and XSD technology [8] and describes the syntactic structure of the document, plus some semantic constraints. Additionally logic was included to reference external files, thus providing a way to build up a complete dataset by aggregating several files over an arbitrary subdirectory structure, a feature not supported by plain XML, although W3C recommends the use of XLink [9].

B. Software integration

As the basic integration platform upon which to integrate the CPACS data format Phoenix Integration's *ModelCenter* was chosen [10]. To enable scientists to use the CPACS data schema within ModelCenter, software engineers at the institute for Simulation and Software Technology developed an integration component suite for ModelCenter as well as helper libraries described below. The architecture of the tool suite can be summarized as follows:
– Each contributing institute or department at DLR installs a server software from Phoenix Integration called *Analysis Server*, running on a physical server machine and offering their workgroups' scientific optimization software as services available inside DLR's network. It can then be accessed from ModelCenter's graphical user interface.
– On any partaking user's desktop computer ModelCenter is installed.
– For the server software a Java component was developed to enable scientists to remotely run any tool or application they might need from within the comfortable project building environment of ModelCenter. This functionality is originally built into ModelCenter, but the plugin contains additional data handling necessary to provide an easy way to exchange and access CPACS data.
– This again is simplified by a set of easy to use Java plugins within the graphical project editor, some of which are described below and in the user handbooks [11] that come along with the component suite.

C. Software libraries support

There are two helper libraries for data manipulation, namely *TIXI* and *TIGL*, being named after the TIVA project for historical reasons.

1) TIVA XML Interface (TIXI): Many applications use input files based on quite simple data structures. Often single floating point or integer numbers are used, and their meaning depends on the exact position of these numbers in the input or output file. More advanced types of these data files are name/value pairs or lists of numbers to reflect vector or matrix data. Based on these requirements it was decided to design a library adhering to an API that supports developers while accessing and archiving their data via a simple interface for operations on the underlying XML structure. On top of the popular and stable XML-library libXML2 (http://www.xmlsoft.org), the C-library *TIXI* was designed to shield application authors from the complexity of XML processing when performing simple tasks like writing an integer or reading a matrix or vector, not unlike the supporting API of the CDF format. In most cases only the XPath to the location inside the XML tree representation and/or the corresponding data is required to read or write data. Language interfaces and example applications for Fortran and Python are also provided, backed up with unit test suites. Fortran is still often used for mathematic calculations, and Python is a very popular and easy to learn scripting language.

2) TIVA Geometry Library (TIGL): Another library for easy processing and querying of geometric data stored inside CPACS data sets was developed and is called *TIVA geometry library*. With *TIGL*, build upon the open source library *OpenCASCADE* (http://www.opencascade.org), it is possible to directly execute geometric functions on fuselage and wing geometries contained in a loaded CPACS dataset in memory, with plans to add functions for other aircraft parts in the future, if need arises (e. g., for engines).

TIGL is written in the C++ programming language and uses the *TIXI* library to load, store and modify CPACS data sets, while leveraging data manipulation of all supported geometry types to build up a 3D-Model of included aircraft in memory as well as querying certain spatial information about the aircraft. The functional range goes from the computation of surface points in Cartesian coordinates up to the export of airplane geometry to different file formats (e. g., IGES [12] and STL (http://www.ennex.com/~fabbers/StL.asp), conversion to Step/Express being under development [7]).

Also, *TIGL* could be used to obtain information about the geometric structure of a plane, for instance how many wings and fuselages the current airplane configuration has, since such a fundamental setting is one of the interesting things examined in some projects. The library provides external interfaces to C and Fortran and could be used to build up new supporting programs. One example is the CPACS TIGL-Viewer, a software written to visually check the correctness of aircraft geometry data contained in a CPACS data set.

3) ModelCenter integration component suite: The title encompasses all Java plugins designed to simplify working with CPACS data from within the ModelCenter integration environment. The following list sums up the most important plugins developed so far, with more to come:

- **Instantiator** helps adding and configuring tools from any server in the network into the current ModelCenter workspace. Without this tool, the user would have in each client project to manually set up IP addresses, folder location and configuration of all the servers, tools and versions available for collaboration
- **Splitter components** help to extract single values from a complete CPACS dataset for connection with ModelCenter's built-in optimization helpers
- **Merger components** complement the splitters by updating single values on CPACS data sets, e. g., to build up a loop over a tool chain (or more complex workflow)
- **Source** enables the user to read in a data set from her filesystem, making use of the TIXI library internally
- **Destination** makes it possible to store computation results to the filesystem. Additionally the intermediate results of each run of an optimizing loop can be logged with this component for later offline analysis
- **Log Viewer** shows local and remote log files, greatly simplifying finding the source of errors

IV. Application in a real world problem solving

As described in [1], the foremost advantage of our approach is the flexibility regarding dynamic presence or absence of data connections between serial and parallel components in a workflow, seldom found in commercial software integration frameworks. Since no new capability comes without a cost, the aforementioned tool suite was developed to simplify working with the new flexibility of transfer and transformation of tree data structures.

To gain useful model starting values, ModelCenter offers a tool called *Parameter study* that allows any combination of parameters values to be tested out. The results are visualized conveniently and may give ideas about local extrema, problem areas and border cases. At DLR it was also observed as good practice to build up smaller toolchains from all existing tools of a desired workflow to gain knowledge about the interdependencies of pairwise tool loops, before integrating them within a larger workflow.

A. Exemplification of a predesign optimization tool chain in aviation

The reduction of local aircraft noise pollution during approach and departure operations is a major goal of ongoing DLR research activities. To reduce ground noise impact, modifications to the aircraft design as well as to the flight procedures are necessary. Obviously, this involves multiple interacting disciplines such as fuselage, wing and engine design, flight mechanics, and noise prediction.

At DLR these disciplines are represented by different institutions and their expert tools. As an example for the framework integration from the TIVA project, we will outline a tool chain for the environmental analysis of flight procedures [13]. Geometric aircraft design, aerodynamic performance, and engine design are required input data for the process. The first tool in the process is TWdat [14]. Therein, engine design is evaluated and a corresponding performance deck is passed on to the next tool. If necessary, effects of the engine installation position on noise radiation are computed with SHADOW [15], using an ray-tracing approach. The modified engine noise characteristics are fed back into the process as CPACS input for the noise prediction. The aircraft is simulated along specified flight segments, calculated with FDS [16] for take-off and approach trajectories. Each flight segment is controlled by user defined parameters. Finally, the individual flight segments are assembled into a complete simulated flight procedure. Aircraft noise and engine emissions are then predicted with the PANAM tool [17]. The output of the tool chain consists of noise levels and isocontour areas for the selected flight procedure. The workflow described above was manually assembled within the graphical user interface of ModelCenter, augmented with the additional components created by the authors to allow for easy CPACS data communication, and can now be executed anytime from everyone inside DLR's local area network. The integration of the tool chain into the presented framework allows for fully automated environmental analyses, and flight segments can be optimized for low-noise operation within the framework. The whole tool chain can be extended and embedded into larger assessments easily.

V. Conclusion and future work

For the time being, our approach to data and software integration already produced working tool chains in several projects. The integration example in this paper shows how easy it is to integrate existing tools with the CPACS data interfaces by using the libraries and components provided by the authors' framework.

Of course there are still challenges to solve not genuinely tackled by the chosen integration platform, e. g., management of data sets larger than a few dozen MiB, user rights management and integration of the CPACS format into other frameworks. The currently employed integration platform *ModelCenter* is to be phased out during coming projects and replaced by the *Reconfigurable Computing Environment* RCE [18]. This again shows the versatility and level of abstraction of our approach and the advantage of a flexible domain-specific data set with the possibility of easy adaption to other environments and integration of new scientific data.

References

[1] A. Bachmann, M. Litz, M. Kunde, and A. Schreiber, "A dynamic data integration approach to build scientific workflow systems," in *Proceedings of the 4th International Conference on Grid and Pervasive Computing*, vol. 1, Geneva, May 4. – 8. 2009.

[2] C. Bochner, "Architekturentwicklung eines erweiterbaren Expertensystems zur Anwenderunterstützung bei flugphysikalischen Simulationen," Master's thesis, Fachhochschule Bonn-Rhein-Sieg, Feb. 2009. [Online]. Available: http://elib.dlr.de/58104 [2009-07-08]

[3] "Common data format." [Online]. Available: http://cdf.gsfc.nasa.gov [2009-07-09]

[4] "Hierarchical data format." [Online]. Available: http://www.hdfgroup.org [2009-07-09]

[5] "Iso 10303-1: Industrial automation systems and integration – product data representation and exchange – part 1: Overview and fundamental principles," I. TC 184/SC4 ISO Standard, 1994.

[6] M. Litz, H.-P. Kersken, and H. Cornelsen, "A software environment for multi-disciplinary simulation in aircraft predesign," in *Advances in engineering software*, 2009, To be published elsewhere.

[7] D. Böhnke, "Dataintegration in preliminary airplane design," Master's thesis, Universität Stuttgart, Jul. 2009, Work in progress, to be published in Sep. 2009.

[8] E. van der Vlist, *XML Schema: The W3C's Object-Oriented Descriptions for XML*, illustrated ed. O'Reilly, 2002.

[9] "Xml Linking Language (XLink) version 1.1," World Wide Web Consortium (W3C), Jun. 2006. [Online]. Available: http://www.w3.org/TR/2006/CR-xlink11-20060328/ [2009-07-08]

[10] "Phoenix integration website." [Online]. Available: http://www.phoenix-int.com [2009-07-09]

[11] A. Bachmann, M. Kunde, and M. Litz, "Analysis Server CPACS integration handbook and ModelCenter integration components documentation," Available upon request to Arne Bachmann (arne.bachmann@dlr.de).

[12] "Initial graphics exchange specification," National Bureau of Standards, Version 4.0, NBSIR 88-3813, 1988.

[13] L. Bertsch, G. Looye, and M. Lummer, "Integration and application of a tool chain for environmental analysis of aircraft flight trajectories," in *The 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Sep., 21 – 24 2009.

[14] F. Deidewig, "Ermittlung der Schadstoffemissionen im Unter- und Überschallflug (Determination of noxious emissions in subsonic and supersonic flight)," German Aerospace Center, Cologne, Germany, DLR Forschungsbericht 98-10, 1998.

[15] M. Lummer, "Maggi-Rubinowicz diffraction correction for ray-tracing calculations of engine noise shielding," in *14th AIAA/CEAS Aeroacoustics Conference*, no. 2008-3050, 2008.

[16] G. Looye, S. Hecker, T. Kier, C. Reschke, and J. Bals, "Multi-disciplinary aircraft model development using object-oriented modelling techniques," in *DGLR Luft- und Raumfahrtkongress*, ser. DGLR-Jahrbuch 2005, Friedrichshafen, Germany, 2005.

[17] L. Bertsch, W. Dobrzynski, and S. Guérin, "Tool development for low-noise aircraft design," in *14th AIAA/CEAS Aeroacoustics Conference*, no. 2008-2995, 2008.

[18] D. Seider, "RCE homepage," 2008. [Online]. Available: http://www.rcenvironment.de [2009-07-08]