The Quest for a Unified Aircraft Dataset Format

Mr Geoff Brian; Mr Michael Young.
Defence Science and Technology Organisation.
506 Lorimer St, Fishermans Bend, 3207.
Victoria, Australia.
Geoff.Brian@defence.gov.au
Michael.Young@defence.gov.au

Dr Daniel Newman; Mr Robert Curtin.
Ball Solutions Group.
Level 3 The Como Centre, 650 Chapel St, South Yarra, 3141.
Victoria, Australia
dnewman@ballsolutions.com
rcurtin@ballsolutions.com

Ms Hilary Keating.
Fortburn Pty Ltd.
23 Rocklea Drive, Port Melbourne, 3207.
Victoria, Australia.

Abstract. In 2002, Air Vehicles Division (AVD) of the Defence Science and Technology Organisation (DSTO) undertook a review of DSTO's fixed-wing aircraft modelling and simulation needs and capabilities, with the aim of improving the quality of technical support provided to Defence aircraft model users. One major recommendation of the review was to improve the encapsulation and management of aircraft model data, such that they may be validated and maintained with available resources, whilst supporting diverse applications ranging from point-performance estimation through to high-fidelity flight dynamic models. The American Institute of Aeronautics and Astronautics (AIAA) is developing a standard for aircraft modelling and simulation, together with a dataset exchange format that meets this standard based on the eXtensible Mark-up Language (XML). This is known as the Dynamic Aerospace Vehicle Exchange Mark-up Language (DAVE-ML), and provides a formal structure for the many data elements that make up a flight model. AVD adopted DAVE-ML as the basis for encapsulating aircraft properties for all of its modelling and simulation applications, and is participating in development of the AIAA standard. Furthermore, AVD intends to use this format not only for exchange, but also natively within each simulation. This has led to the development of an interface application between the aircraft datasets and modelling codes, which abstracts the content of the dataset in the form required by each model. DSTO's initial usage of DAVE-ML, its benefit to modelling and simulation applications, and the DSTO interface application will be presented in this paper. Results of this approach to-date show significant advantages over the data structures and interfaces used previously, in terms of completeness and flexibility in a range of flight model applications, and for the computational performance achieved.

1. INTRODUCTION

The Australian Defence Force (ADF) is a significant user of aircraft modelling and simulation products and services to support aircraft acquisition projects, through-life service support and training of personnel. In supporting the ADF's requirements, DSTO is both a major developer and user of these aircraft models and simulations.

The development and use of aircraft models and simulations within DSTO has grown rapidly during the past decade, supporting diverse areas including flight performance and handling; structural life prediction; sensor performance; operational analysis and tactics development, operator performance analysis and training. The diversity of these fields has resulted in an equally diverse set of modelling applications representing aircraft of interest. A number of these models have become difficult to support due to maintenance and adaptability constraints. Further, uncertainty has been expressed with the appropriateness of model fidelity for the different end use applications and the consistency of the aircraft representations. As a result, in 2002, DSTO Air Vehicles Division (AVD) undertook a review of fixed-wing aircraft modelling and simulation needs and capabilities throughout the DSTO with the aim of improving the quality of technical support provided to Defence aircraft model users. A broad set of recommendations arose from the survey, which are presented in [7], [8], of which the unification of aircraft datasets for the various modelling applications was rated as a high priority.

The challenge was to develop a dataset structure that would encapsulate data characterising a candidate aircraft in a single framework, which could be used by aircraft models ranging from simple point-performance models (used for flight, sensor and operator performance analysis) to high-fidelity flight dynamic and simulator training models. In addition, the intention was to minimise the constraints on how the aircraft data were encapsulated within the datasets so they closely reflected their source. Furthermore, data validation and the ease of maintenance were considerations.

A review of dataset encapsulation techniques found that the American Institute of Aeronautics and Astronautics (AIAA) is developing a standard for aircraft modelling and simulation, together with a dataset exchange format based on the eXtensible Mark-up Language (XML), which meets the standard [1], [4]. The format is known as the Dynamic Aerospace Vehicle Exchange Mark-up Language (DAVE-ML), and provides a formal structure for the many data elements that make up a flight model [5]. DSTO found that DAVE-ML would suitably encapsulate aircraft data meeting the aforementioned requirements for the unified format [3]. As a result, AVD adopted DAVE-ML as the basis for encapsulating aircraft properties for all of its modelling and simulation applications, and is participating in development of the AIAA standard. This has led to the development of an interface application between the aircraft datasets and modelling codes, known as Janus, which abstracts the content of the dataset in the form required by each model.

DSTO's initial usage of DAVE-ML and its benefit to modelling and simulation applications will be discussed in this paper, together with the development of the Janus interface application.

2. DYNAMIC AEROSPACE VEHICLE EXCHANGE MARKUP LANGUAGE (DAVE-ML)

The eXtensible Markup Language (XML) is defined by a World-Wide Web Consortium (W3C) sponsored standard, and is used to structure documents using simple, human-readable tags to encapsulate elements of information. Due to its flexibility, XML has found application and support in a diverse range of industries, including financial services, real estate and publishing.

The Dynamic Aerospace Vehicle Exchange Markup Language (DAVE-ML) is an application of XML being developed by an informal team of members of the American Institute of Aeronautics and Astronautics (AIAA), in particular staff from NASA Langley Research Centre and Science Applications International Corporation (SAIC). This team has been working together since 2002 with the objective of developing "an XML application to encode complete flight vehicle dynamic models in a facility- and language-independent, consistent way, to expedite model exchange and validation between different simulation facilities and tools" [6].

The rules regarding the content of a particular XML application may be defined in an associated Document Type Definition (DTD) or XML Schema. The DAVE-ML syntax is defined in the DTD DAVEfunc.dtd, which is available from the XML.org Registry <http://www.xml.org>, and the DAVE-ML web site <http://daveml.nasa.gov>.

The DAVE-ML format includes a header section, variable definitions, breakpoint definitions and functions. The header contains information about the file, including its author, version number, creation date, source references and a modification history. Defined variables may be constants, inputs to one or more functions, dataset outputs, or internal parameters. Functions may include or reference look-up tables, which may consist of sets of data points, or multi-dimensional gridded tables. For the gridded table definitions, child elements define the dependent and independent variables and refer to the breakpoint sets defined earlier. Information about the data source and confidence may be included, and attributes may be used to record instructions for the extrapolation and limiting of data in each table, in each dimension.

An example DAVE-ML compliant XML file is shown in Figure 1.

In addition to the application-specific tags it provides, DAVE-ML invokes MathML. MathML is an application of XML that allows the use of two sets of mark-up tags to describe mathematical elements: One set describes presentation (ie, the way that the data, symbols or equations should be rendered in a web page presentation); the other set describes content (ie, the relationship of each of the mathematical elements to each other). In DAVE-ML, content MathML may be used to describe relationships between variables and function tables.

Although DAVE-ML was intended as an exchange format, AVD decided to adopt it as its native dataset format for its aircraft modelling and simulation applications. This takes greater advantage of XML’s benefits, namely its flexibility and supportability over the longer term. However, as DAVE-ML is simply a format for text documents, this decision has necessitated the development of a programming interface to enable the datasets to be transformed into functions accessible by applications requiring the data.

```xml
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE DAVEfunc SYSTEM "DAVEfunc.dtd">
<DAVEfunc>
     <fileHeader>
          <author name="Mary Smith" org="DSTO AVD"/>
          <fileCreationDate date="2005-05-09"/>
          <fileVersion>1.0</fileVersion>
          <description>This is an example DAVE-ML compliant XML dataset</description>
          <reference refID="R1" author="Janes" title="All the World's Aircraft" date="2005"/>
          <modificationRecord modID="A"><author name="John Citizen" org="DSTO AVD"/>
               <description>Added drag coefficient output variable and function</description>
          </modificationRecord>
     </fileHeader>
     <!-- ========= Input variables ========= -->
     <variableDef name="Angle-of-Attack" varID="angleOfAttack" units="deg">
          <description>Angle-of-attack</description>
     </variableDef>
     ...
     <!-- ========= Constant variables ========= -->
     <variableDef name="Wing Area" varID="wingArea" units="ft2" initialValue="150">
          <description>Wing reference area</description>
     </variableDef>
     ...
     <!-- ========= Output variables ========= -->
     <variableDef name="Lift Coefficient" varID="liftCoefficient" units="ND">
          <description>Lift coefficient</description>
          <isOutput/>
     </variableDef>
     <variableDef name="Lift" varID="lift" units="lbf">
          <description>Lift force</description>
          <calculation>
               <math>
                    <apply>
                         <times/>
                         <cn>0.5</cn><ci>rho</ci><ci>liftCoefficient</ci>
                         <apply><power/><ci>velocity</ci><cn>2</cn></apply>
                         <ci>wingArea</ci>
                    </apply>
               </math>
          </calculation>
          <isOutput/>
     </variableDef>
     ...
     <!-- ========= Breakpoint values ========= -->
     <breakpointDef name="Angle-of-Attack Breakpoints" bpID="angleOfAttackBP" units="deg">
          <bpVals>0.0, 5.0, 10.0, 15.0, 25.0, 30.0, 40.0, 55.0</bpVals>
     </breakpointDef>
     ...
     <!-- ========= Functions ========= -->
     <function name="Lift Coefficient Function">
          <description>Lift coefficient function</description>
          <provenance><author name="Mary Smith" org="DSTO AVD"/>
               <functionCreationDate date="2005-05-09"/>
               <documentRef docID="R1"/>
          </provenance>
          <independentVarRef varID="angleOfAttack" min="0.0" max="70.0" extrapolate="both"/>
          <independentVarRef varID="mach" min="0.0" max="2.0" extrapolate="min"/>
          <dependentVarRef varID="liftCoefficient"/>
          <functionDefn>
               <griddedTableDef>
                    <breakpointRefs>
                         <bpRef bpID="angleOfAttackBP"/>
                         <bpRef bpID="machBP"/>
                    </breakpointRefs>
                    <dataTable>
                              0.00, 0.45, 0.92, 1.33, 1.93, 2.14, 2.39, 2.38,...
                    </dataTable>
               </griddedTableDef>
          </functionDefn>
     </function>
     ...
</DAVEfunc>
```

Header section

Variable definitions

Example MathML

Breakpoint definitions

Function definitions

Figure 1: Example DAVE-ML compliant XML dataset

3. DEVELOPMENT OF AN INTERFACE CLASS FOR DAVE-ML DATASETS

3.1 Flight Model Data Sources

Data underlying a flight model can occur in many forms and have many origins, including flight test, wind tunnel test, Computational Fluid Dynamics (CFD) at varying levels of fidelity, and empirical or semi-empirical models based on historical data. The DAVE-ML data structure is able to accommodate all of these sources of data, and to associate measures of fidelity with each data element. It can also include descriptive characterisation of the data. Given the many forms that flight model data can take, it is apparent that a model developer might easily be swamped by data handling issues, to the detriment of actual model development and subsequent model output quality. The capability and flexibility of the data structure adopted by DSTO required an equally capable and flexible interface between data and model, without excessive model complexity.

3.2 Purpose of Interface Class

The development of the Janus class [9] was primarily motivated by the requirement, dictated by the multiplicity of raw data sources, to abstract the aircraft flight model data from the model itself.

3.3 Implementation of Interface Class

The Janus interface class seeks to avoid model data-handling difficulties by presenting a common Application Programming Interface (API) for specifying a flight model state, and for obtaining current values for dataset-based variables dependent on that state. An instance of the C++ interface class loads the content of a DAVE-ML compliant XML dataset to a Document Object Model (DOM) within the instance, then parses the relevant components of the DOM to set up numerical structures corresponding to the dataset. It maintains an array of the current values of all state variables defined within the dataset, which are used to compute dependent variable values and return the results through the data-independent interface whenever requested by the calling program.

The DAVE-ML dataset contains not only data, but also explicit instructions for processing that data to obtain dependent variable values. The current implementation of the Janus class handles data in the following forms:

1. Gridded data, including non-uniform grids, up to 32 input degrees of freedom (DoF) (typical aero-propulsive models use between 2 and 5 input DoF for most outputs);

2. Ungridded data, such as from wind tunnel or flight test, up to 32 input DoF; and

3. Equation-based data, using all common arithmetic, trigonometric and logical functions, including piecewise-defined equations.

For the tabulated data forms, interpolation can be specified as discrete, linear, or polynomial, and extrapolation can be controlled in each direction for each degree of freedom. Ungridded data interpolation is based on Delaunay tessellation by the Quickhull algorithm, performed during instantiation using the open-source Qhull library [2]. All computations have been programmed to maximise data output rate while retaining model structure flexibility.

Performance testing conducted by AVD demonstrates that the Janus data-handling interface has the potential to run models at rates well in excess of real time without requiring exotic computational hardware. This has positive implications for flight model applications involving statistics-based prediction. Typical performance levels on representative aero-propulsive coefficient evaluations (performed on a 750 MHz PC, using gcc v3.3.5 under Linux, using linear interpolation for tabulated data) are shown in Table 1.

Table 1: Janus interface computational performance

Computation Type	Output Rate (sec^{-1})
Gridded, 1 DoF, 8 breakpoints	1.70E+006
Gridded, 2 DoF, 8 * 5 breakpoints	9.13E+005
Gridded, 3 DoF, 8 * 5 * 4 breakpoints	6.09E+005
Gridded, 4 DoF, 8 * 5 * 4 * 3 breakpoints	2.70E+005
Ungridded, 1 DoF, 8 data points	4.72E+005
Ungridded, 2 DoF, 40 data points	6.07E+004
MathML, 1 DoF, 6^{th} order polynomial	5.35E+005

Since the Janus instance performs all data handling in accordance with the instructions in the XML dataset, the calling program which implements the model can be entirely flight vehicle independent. Providing the dataset is structured to return appropriate driving function values for the model, the details of flight vehicle configuration become irrelevant to the model builder. This applies even to such fundamental characteristics as numbers of engines. A new aircraft (or any vehicle obeying Newtonian mechanics at the level of model validity required) can thus be modelled without even recompiling the model code base.

Janus interface capabilities demonstrated to date have the potential to dramatically increase the range of vehicles which can be modelled, while simultaneously reducing the effort required to maintain and support the model code base. DSTO staff can concentrate on quantifying the characteristics of flight vehicles of interest, rather than manipulating model code to handle different data formats.

4. BENEFITS FOR FLIGHT MODELLING AND SIMULATION APPLICATIONS WITHIN DSTO

A key benefit of XML for modelling applications is the ability to exchange data between users and across platforms. Within DSTO, AVD supplies aircraft data to a number of other Divisions for use in a variety of applications. Often this data must be converted manually to the local native format, and may be supplemented to support local applications. In some cases, data for the same aircraft are sourced separately through the third-party simulation supplier. Standardisation on a UNICODE text-based dataset format, with the clearly defined yet flexible structure of DAVE-ML, allows the reliable dissemination of validated aircraft model data across Defence, eliminating the need for duplicate, and inevitably diverging datasets. The DAVE-ML structure, combined with automated validation techniques within Janus will greatly reduce the risk of misinterpretation by the end user or by their local simulation code.

The Janus API was developed to allow the use of DAVE-ML natively by DSTO models; however, standardised methods and tools do exist for the conversion of XML files from one format (DTD or Schema) to another, if required for legacy systems. The intention is to use these methods to automate the conversion process, so that the source data is always contained in a DAVE-ML file, to which changes are made and managed.

The encapsulation of data processing instructions and other supporting information within DAVE-ML has already yielded a number of significant benefits for some DSTO applications. The ability to specify different interpolation methods and boundary handling for each dimension of a dataset has resulted in improved accuracy for high fidelity simulations, and less time and risk in pre-processing or "massaging" data that are limited in scope. Previously, when disseminating aircraft datasets within DSTO, AVD have provided caveats and limitations of the data in separate documentation. However, awareness of these limitations is often lost over time, particularly where the dataset is translated to new formats and modified. DAVE-ML is now being used to ensure that this link is not broken. Another feature of DAVE-ML that DSTO plans to exploit more in future is the ability to record validation information, including check case data, which may be used for acceptance testing in a new model environment.

The first DSTO applications to benefit from DAVE-ML and Janus implementation are the two multi-aircraft modelling architectures at AVD, named Amiel and Merlin. Amiel is an architecture in which all locally developed flight dynamic models are being maintained, including some migrated legacy models. This architecture is highly modular, standardising generic features such as environmental models, equations of motion and common aspects of aircraft aerodynamics and systems. Amiel modules are selected, sometimes from optional levels of fidelity, and packaged to produce a tailored application for the user. Aircraft specific features are implemented at the lowest level, where possible within one of several DAVE-ML files that mirror the object structure (aerodynamics, propulsion, etc, as identified by a master DAVE-ML file). Merlin, a successor to the earlier DSTO Aircraft Performance Estimation Software (DAPES) tool, is designed for analysing fixed-wing aircraft performance. This uses a pseudo-static model (ie, no numerical time integration), with a very different functional approach to that of Amiel. Merlin code is completely generic, with aircraft defined purely by their datasets.

Both Amiel and Merlin require datasets for a variety of aircraft. Traditionally, separate and dissimilar datasets have been maintained for the performance and dynamic models. Performance datasets are typically a subset of the data for a higher fidelity model, but with aerodynamic data based on trimmed conditions across the flight envelope. Also, performance datasets have been necessarily identical in structure, whereas dynamic model datasets have been unique and matched to aircraft-specific code. Using DAVE-ML, Merlin will be able to extract only the data it requires, which will be available in the same form from all datasets. However, that common form at the interface may be the result of MathML operations on dissimilar source data structures for each aircraft. The same DAVE-ML files will contain additional data that is only relevant to dynamic models, but which will not be accessed by, or impact the Merlin code. Again, MathML may be used to specify the aircraft-specific build-up of dataset components to a common level, eg, total forces and moments.

This approach will greatly ease the burden of dataset production and maintenance. The ubiquity of XML-based formats ensures that these tasks are well supported by both commercial-off-the-shelf and open-source tools. Encapsulation of the data allows different user interfaces for editing and viewing the data, just as HTML editors allow multiple views of a web document. The unification of performance and dynamic model datasets also allows far greater reuse of code in the organisation, while providing a more flexible and reliable data specification for all end applications.

5. CONCLUSIONS AND RECOMMENDATIONS

The implementation of a unified aircraft dataset format is a non-trivial activity and, whilst the improved capability achieved to date is very encouraging, the work is not complete. However, the architecture of the DAVE-ML format, together with the AIAA standard for aircraft modelling and simulation, provides a viable framework to achieve this goal.

DSTO will continue its collaboration in developing the AIAA aircraft modelling and simulation standard, together with the DAVE-ML dataset language. Future developments will be reflected in the Janus interface class.

The use of DAVE-ML, together with the Janus interface, promises to significantly improve the development, validation and on-going management of fixed-wing aircraft flight models within DSTO. Furthermore, the consistency of the aircraft representations used by the diverse set of modelling applications will increase and, therefore, improve the quality and efficiency of technical support provided by DSTO to Defence in the future.

REFERENCES

1. Anon., (2004) Dynamic Aerospace Vehicle Exchange Markup Language (DAVE-ML) Reference, Version 1.7b1, AIAA Simulation Standards Working Group.

2. Barber, B., “Qhull Manual (online version),” The Geometry Center, Minneapolis MN, December 2003, <http://www.qhull.org> (cited 1 February 2005).

3. Curtin, R.J. (2003) “Application of XML to Aircraft Performance Datasets”, Ball Solutions Group Report 1234.004.

4. Jackson, E.B. & Hildreth B.L. (2002) “Flight Dynamic Model Exchange Using XML”, AIAA 2002-4482, AIAA Modeling and Simulation Technologies Conference, Monterey, CA, August 2002.

5. Jackson, E.B.; Hildreth B.L.; York B.W. & Cleveland W.B. (2004) “Evaluation of a Candidate Flight Dynamics Model Simulation Standard Exchange Format”, AIAA 2004-5038, AIAA Modeling and Simulation Technologies Conference, Providence, Rhode Island, August 2004.

6. Jackson, E.B., “Flight Dynamic Model Exchange using XML” <http://daveml.nasa.gov> (cited 21 January, 2005).

7. Newman, D.M. (2004) “Efficient Development and Use of Aircraft Flight Models – Survey of DSTO Usage”, Ball Solutions Group Report 1234.002.

8. Newman, D.M.; Curtin, R.J. & Brian, G.J. (2004) “DSTO Flight Model Usage – Present and Future”, SimTecT 2004, Canberra, 24th-27th May 2004.

9. Newman, D.M. (2005) “The Janus C++ Library - An Interface Class For DAVE-ML Compliant XML-Based Flight Model Datasets”, Ball Solutions Group Report 31495.002.