



AIAA 2002-4482

## Flight Dynamic Model Exchange using XML

E. Bruce Jackson  
NASA Langley Research Center  
Hampton, Virginia

Bruce L. Hildreth  
SAIC  
Patuxent River, Maryland

### **Modeling and Simulation Conference**

5-8 August 2002  
Monterey, California

## FLIGHT DYNAMIC MODEL EXCHANGE USING XML

E. Bruce Jackson<sup>\*</sup> *NASA Langley Research Center, Hampton, VA*  
Bruce L. Hildreth<sup>†</sup> *SAIC, Lexington Park, MD*

### Abstract

The AIAA Modeling and Simulation Technical Committee has worked for several years to develop a standard by which the information needed to develop physics-based models of aircraft can be specified. The purpose of this standard is to provide a well-defined set of information, definitions, data tables and axis systems so that cooperating organizations can transfer a model from one simulation facility to another with maximum efficiency.

This paper proposes using an application of the eXtensible Markup Language (XML) to implement the AIAA simulation standard. The motivation and justification for using a standard such as XML is discussed. Necessary data elements to be supported are outlined. An example of an aerodynamic model as an XML file is given. This example includes definition of independent and dependent variables for function tables, definition of key variables used to define the model, and axis systems used.

The final steps necessary for implementation of the standard are presented. Software to take an XML-defined model and import/export it to/from a given simulation facility is discussed, but not demonstrated. That would be the next step in final implementation of standards for physics-based aircraft dynamic models.

### Introduction

Establishment of the Internet and rapid adoption of the World-Wide Web has led to the definition of standards for the electronic exchange of many types of information. A notable exception to this has been the reluctance of the flight simulation community to adopt any standards on model exchange formats. Several earlier attempts by the AIAA Flight Simulation Technical Committee and others have met with lukewarm reception. It now appears that other science disciplines (including mathematics, chemistry and biology) are moving toward data exchange standards, based in part on a Web technology known as Extensible Markup Language or XML<sup>1</sup>. These exchange standards allow rapid dissemination of molecular models, DNA sequences, and mathematical descriptions. This paper proposes some first steps for the flight dynamics community to move in this same direction.

### Motivation

New aircraft development projects typically involve more than one major contractor, several subcontractors and often government laboratories. These multiple partners generally exchange flight dynamic models on a recurring basis as the vehicle is developed. In the authors' experience, the initial sharing of a flight dynamic model requires extensive manual involvement to modify ("re-host") the model to fit into and operate within the architecture of each partner's simulation tool, whether piloted or batch. The first exchange is usually quite tedious and involved, taking months of time and staff-years of effort. Subsequent exchanges are less difficult but still require considerable time and effort to install and verify at each participating facility.

After the vehicle is developed, development of training simulations requires similar energy in putting together flight dynamic models that are compatible with the training simulation vendor's architecture.

There is also an "aftermarket" for flight dynamic models: the home gaming and simulation community. This group is getting quite proficient at sharing increasingly complex simulations of popular aircraft, and has moved toward open software standards and exchange mechanisms. While these efforts are not of the fidelity of typical commercial and military simulation models, considerable innovation has been shown in sharing vehicle models. In fact, an early use of XML in this regard has been demonstrated in one open-source flight simulation development project.<sup>2</sup>

At the same time that vehicle model exchanges are becoming more prevalent, the complexity and fidelity of the flight dynamic models are increasing. For example, a recent NASA project involving multiple control surfaces and their interactions ran up against the ANSI FORTRAN compiler limitation of 7 dimensions in an array. This makes the re-hosting task more tedious and time-consuming to assure the nuances of hinge moment and actuator models, control surface aerodynamic interactions, and aeroservoelastic coupling are faithfully captured.

The emergence of the World-Wide Web has led to several innovative information exchange standards. The well-known Hypertext Markup Language (HTML), based on the Specialized General Markup Language (SGML) specification, is used to exchange mostly textual and graphical information via well-known "web

---

<sup>\*</sup> Senior Research Engineer, Dynamics & Control Branch, Airborne Systems Competency; Senior Member, AIAA

<sup>†</sup> Vice-President/Division Manager; Senior Member, AIAA

browser” applications. A more recent development known as the Extensible Markup Language (XML), also based on SGML, has emerged as a potential adjunct to HTML to provide for rapid exchange of more specialized information. XML is a text-based document framework, designed by the World-Wide Web Consortium (W3C), in which many types of data can be embedded, transmitted, and understood. XML allows the encapsulation of definitions, documentation, references, models, and data within a single computer file. Specialized “flavors” of XML are being developed, for example, to exchange chemical formulae such as the definition of complex molecules. This subset, known as Chemical Markup Language (CML), allows researchers to exchange complete models of chemical compounds. With the proper plug-in for web browsers, a CML file allows 3-D visualization of the molecule's structure, as well as supporting import of the model data into chemical software applications.

Clearly the time has arrived for similar capabilities regarding flight dynamic models. A subset of XML that is programming language and facility-neutral could be developed to allow interchange of unaugmented flight vehicle aerodynamics, propulsion, and landing gear models – the basic components of all high-fidelity flight simulation programs. The same data package could include documentation and check case data to ensure the understanding and verification of the model when delivered. Re-hosting of models, once the model package is available through the Internet (encrypted as necessary) or on suitable digital media, could be extremely rapid once XML-savvy import scripts or even “flight simulation player” applications are developed.

Part of the reluctance to adopt simulation standards is due, in part, to the preference for in-house developed simulation architectures. Through the development and implementation of an exchange standard, facilities are free to retain their well-known and trusted simulation hardware and software infrastructures with perhaps minor revisions. The major effort required will be to develop software to export existing and import new flight dynamic model packages into/from the XML-based model standard.

### Background

#### Dissimilar simulation architectures

Simulation facilities associated with each of the major airframe manufacturers and government laboratories have been developed more or less independently as computer technology and software methods improved. Some cross-pollination has occurred, but most simulation facilities use their own methods of table lookup algorithms, programming languages, and variable naming conventions.<sup>3</sup>

#### No official standards, some conventions

Luckily, most simulation facilities adhere to a right-hand rule for X, Y, and Z body axes. Aside from that, however, differences arise. Many simulation models assume the X-body axis is positive going forward through the nose of the vehicle, but there exists at least one manufacturer that adheres to the “configurator's axis system” where X is positive out the right wing of the vehicle. There is not even consensus of what to call angle of attack (usually the Greek symbol  $\alpha$ ) or even how to spell it: the authors have seen ALPHA, ALFA, ALPHAD, and ALPDEG used in FORTRAN simulations. Further confusion exists over whether the aerodynamic data is resolved into body- or wind-axes or whether aerodynamic moments generated by the aerodynamic model should be about the current center-of-gravity or some fixed reference point.

#### 1992 ANSI/AIAA Recommended Practice for axes & sign convention

In 1992 the AIAA, in cooperation with the American National Standards Institute, published a Recommended Practice (RP)<sup>4</sup> for use in specifying vehicle model axis systems and sign conventions. It also provided mathematical notation standards for various quantities, including the designation of  $\alpha$  for angle-of-attack. Unfortunately, the RP did not suggest FORTRAN variable names to use, which would have been a big step forward (if somewhat dated with the appearance of other programming languages).

#### 1998 AIAA Modeling and Simulation TC draft standard

In the mid 1990's the Modeling and Simulation Technical Committee voted to support the development of a flight mechanics standard. The standard was designed to be for the interchange of information between simulation facilities (the subject of this paper). The first paper on the subject, reference 5, discussed how a standard should be developed. It discussed general software standards that could be used to apply to flight mechanics simulation and the process for starting with general standards and proceeding to more specific standards, potentially resulting in standard software modules for common flight simulation tools or functions. The committee then proceeded to develop a standard that would satisfy the initial requirements of reference 5. Reference 6 then presented the basis of the standard to allow the simulation community to provide comment and feedback. At the present time, the standard is ready for trial implementation and use. This paper reports upon the potential cost savings to the flight simulation community that would accrue from adopting a flight dynamic model exchange standard.

In the future, this standard could logically be incorporated as part of the Synthetic Environment Data Representation and Interchange Specification (SEDRIS) as an expansion of simulation environmental standards into the flight dynamics domain. Additionally

or alternatively, AIAA is a member of the American National Standards Institute (ANSI) and can submit this standard to ANSI for issuance as an AIAA/ANSI standard. It also may be submitted to IEEE for consideration as an IEEE standard. Subsequent to that, it should be submitted to the International Organization for Standardization (ISO) under the sponsorship of AIAA or IEEE.

#### Numerous DoD proposals

Several attempts by the Department of Defense to modularize simulations have been proposed; only a few have been adopted. These proposals went so far as to specify how many and what type of simulation processor to use (e.g. MODSIM). These proposals have met with limited success.

#### Financial justification for a standard

While it is very difficult to determine the cost savings that might result from a standard, this paper attempts to quantify some of them. At various points in their careers the authors have been involved in taking aerodynamic simulations of tactical aircraft from outside sources and converting / re-hosting them to other simulation architectures, both at the NAWCAD Aircraft Simulation Division's Manned Flight Simulator, and for internal use at SAIC and NASA. Based on the authors' experience, it takes an average of one staff-year to re-host a model from one organization's architecture to another. Certainly this task has been done in as little as a couple staff-months to as many as several staff-years, but one staff-year might be a reasonable average. In addition, other simulation engineers from NASA, government, and industry were polled and their responses (3 in total) indicated that 1.5 staff-years was the average with a low of 0.5 and a high of 8 for re-hosting software. It should be clear that this re-host is for the flight dynamics portion of the model, including aerodynamics, flight controls, inertial characteristics, propulsion, and occasionally landing gear models. This level of effort does not include re-hosting a complete avionics and weapon system and includes only minimal validation. It includes only software and does not include any hardware integration.

In the authors' opinion, the largest economic savings from a standard would arise from the consistency of the models after re-hosting. A standard would reduce the number of defects in models shared by several facilities. The biggest cost in using shared models is the continuing maintenance required to incorporate updates between the participating facilities, validating these updates, and then conducting training or research on models with defects that are found later, requiring the training to be repeated, resulting in negative training, or resulting in invalid research results.

Additional benefit would be a reduction in the schedule for a project utilizing a re-hosted simulation model; an order of magnitude or more reduction in the time

required to perform the re-host may be possible, allowing project schedules to shrink.

An illustrative example of the need for this standard is given for the case of an actual tactical aircraft simulation. This aircraft simulation is used at more than 20 locations and includes more than 59 separate simulators. 31 of these simulators are used for pilot training and the rest are used for research, development, test and evaluation (RDT&E).

Table 1 presents a model of the cost savings that may result from a standard. In theory, all simulators should use the "latest and greatest" simulation for that aircraft. However, the reality is that the flight dynamic models at each installation have diverged considerably since they were distributed. Each individual facility has detected defects and fixed them. In addition, various users have made improvements to the simulation, either in envelope covered, the fidelity of the simulation, or execution speed. Likewise, certain upgrades of the flight control system have been made that are incorporated in some simulations and not others.

The financial impact of not having a standard that allows all these sites to work as a team, maintaining and upgrading the simulation, is in the negative (incorrect) training that results from simulation with sub-optimal fidelity (or outright errors), the research differences that results from different models being used, and the time spent by different facilities all finding and fixing the same defects independently. If a standard existed to exchange simulation models easily, the model updates and corrections would be much more widely shared in a very cost effective manner, resulting in all the simulations having better fidelity, more consistent results, and less negative training at less total cost.

The authors suggest a model for estimating the potential savings from adopting an exchange standard. The model includes three saving components:

- 1)savings in maintaining the simulation models,
- 2)savings due to less bad training or research time,
- 3)savings from improved productivity throughout the suite of 59 type-specific simulators

The input parameters are very conservative estimates and are based on an actual simulator family of a tactical military aircraft. The savings created by the standard are the authors' own opinion and not supported by hard data, but are based on experience in the aerospace industry.

The average trainer acquisition costs were estimated and based on expected lifetime of the simulator (without refurbishment). The trainer utilization was assumed to be high and from this the average cost of the simulation per hour was conservatively computed. The labor estimates required to maintain the simulator

**Table 1. An Analysis of Savings from a Flight Dynamics Model Standard**

Input Data	Pilot Training Simulators	Desktop RDT&E Simulators		
Number of locations	4	16		
A/C models (all one type)	4	4		
Trainer Models	6	n/a		
Number of simulators	31	28		
Utilization per yr per simulator, hr	4160	500		
Amort. of device, \$/hr	601	60		
Maint and operation, \$/hr	445	300		
Total operational cost, \$/hr	1,046	360		
Cost to implement one model change	\$30,000	\$30,000		
			<b>TOTAL IMPACT</b>	<b>SAVINGS</b>
<b>1) Total Cost to Implement Model Changes Without a Simulation Standard</b>				<b>Savings Factor</b>
Number of Model changes Per Year	2	1		
Total Cost of Changes Per Year	\$360,000	\$480,000	<b>\$840,000</b>	<b>0.5</b>
<b>2) Cost of Lost Trainer or Research Time</b>				
Lost hrs per trainer due to model errors and the time to fix them	100	50		
Total Cost of Lost Time per Year	\$3,241,587	\$504,000	<b>\$3,745,587</b>	<b>0.5</b>
			<b>Sub Total</b>	<b>\$2,292,793</b>
<b>3) Improved Trainer Community Productivity</b>				<b>2.0</b>
				<b>\$4,585,587</b>
			<b>TOTAL ANNUAL POTENTIAL SAVINGS FOR ONE A/C TYPE</b>	<b>\$6,878,380</b>

are minimal; for the trainers, one operator and one maintenance technician per shift were assumed. For the research simulations, only one person was estimated to maintain and operate the simulator and generate results. Cost of the actual aircrew being trained or flying the research simulations was not included. The research simulators were considered to be software-only simulations, hence the conservatively low acquisition costs.

Based on these assumptions, using the estimate of the actual number of trainers, the cost per year of making flight dynamics model changes to the trainer and research communities is conservatively estimated to be \$840K ([6 trainers × 2 changes per year × \$30,000 per change] + [16 research sims × 1 change per year × \$30,000 per change]). The \$30,00 per change estimate is based on one change taking two staff-months to administer, implement, and validate. Additionally, a conservative estimate of the hours lost in research (50 hrs. per simulator) or trainer time lost (100 hrs. per simulator) due to the defects in the flight dynamics model is \$3.2M for trainers and \$0.5M for the research simulations. Therefore, if we estimate that the standard reduces these costs by a factor of 50%, then the community would save \$2.3M (the sub-total in the

table) a year, just in time lost to defects and the cost of making changes.

However, the authors' position here is that the greatest losses in not having a simulation standard for flight mechanics models is the reduced interchange and cooperation between all the simulator sites (59 simulators in 20 locations in the case study) that result from the barriers to exchanging information. The authors have attempted to quantify this, but we could come to no analytical method for doing so. Therefore the authors will boldly state that the cost of the reduced productivity of the community is twice that of the lost time and cost of maintenance discussed above. This results in the additional savings of a standard at \$4.6M per year. While the reader may dispute that figure it is almost impossible to reduce the \$2.3M per year estimate. Therefore the range of savings may be safely stated to be between \$2.3 and \$6.9M a year or more. It must be noted that this savings is only for one aircraft type. The impact of a standard across the whole aircraft simulation community would be many times this.

This is also the reason it is difficult to get organizations to fund a standard. The \$6.9M savings is spread across

59 simulators and 20 sites. The saving per simulator is relatively small (\$117K). Therefore it is difficult to convince any one location or simulator to pay for the development of the standard. The cost savings only becomes apparent when the whole community for one type of aircraft simulator is examined.

### XML Overview

#### Extensible Markup Language

The Extensible Markup Language (XML) specification was developed to facilitate transfer of information in a structured way through electronic means. The specification supports development of specialized mark-up tags (found in angle brackets <>) that provide both structure (syntax) and expectations of the content to a special-purpose digital document.

#### Similarity to HTML

Extensible Markup Language is quite similar to the familiar Hypertext Markup Language (HTML), which is the familiar language for Web content. Both HTML and XML adhere to the Specialized General Markup Language (SGML) standard. However, HTML is aimed at mostly text-based content with embedded graphics; XML is more generalized and not necessarily intended for parsing by Web browsers. XML addresses some structural shortcomings of HTML (such as not requiring paired tags). A key aspect of the XML standard is to make the content human-readable in a text editor.

In a manner similar to how a Web browser interprets and displays information in an HTML document, specialized XML browsers are being developed to “display” and allow interaction with other types of information. As mentioned previously, the chemical engineering profession is learning to use CML, an XML “flavor,” to display, modify, and share chemical compound models. Another XML application is the development of MathML which, either through plug-ins for existing browsers or native capability of enhanced browsers, allows more accurate depiction of mathematical expressions on Web pages. This is possible because HTML documents can embed XML fragments, and vice-versa. These MathML expressions print out at full resolution (unlike current-generation Web pages in which math expressions are low-resolution images) and actually allow interaction with the mathematical formulas within the Web browser.

More importantly, the MathML-encoded expressions convey both display (presentation) and mathematical relationships (context). Thus, MathML information can be used for both documentation as well as symbolic manipulation and numerical calculations.

#### Additional XML features

XML allows inclusion of a rich set of symbolic and international symbols by fully supporting the

UNICODE standard. Thus, Greek symbols can be used as part of the embedded documentation. Other types of data, including links to other on-line documents, are supported. Thus, the build-up equations for lift (for example) could be encapsulated along with the function data, provenance information, confidence bounds, and links to other documentation and references.

A key to XML is that XML-capable browsers and XML processing applications must ignore types of data which they don't understand, while handling data that is recognized. Thus a browser plug-in could be developed that displays the aero data in an interactive graphical environment, along with textual descriptions of the database, while another stand-alone application would allow manipulation of the buildup equations without corrupting the data tables themselves.

#### Applying XML to Flight Dynamic Models

##### Completely self-contained dynamic model "object"

A key to making use of XML for flight models is in developing the syntax and rules (the document type definition [DTD] or data dictionary) that must be present in an XML application. It would be desirable to include all the elements necessary to specify and validate a flight dynamic model, including (but not limited to) mathematical models of vehicle inertias, aerodynamic characteristics, propulsion systems, and landing/arresting gear models. These may be in a single document or split into several associated XML specifications. Also included in the flight model XML document could be vehicle physical characteristics, for example a description of the appearance of the vehicle using a solid geometry model description. This would allow for a real-time simulation of the vehicle to be built entirely from the data contained in a set of XML documents.

Setting up the necessary specification should be overseen by a group of people under the auspices of the AIAA. This would avoid simultaneous development of competing XML-based specifications for similar applications, as has happened in the area of genetic research into the human genome – several competing XML-based specifications are in use.

##### Elements of the specification

The specification for the digital aerospace vehicle model exchange (DAVE) markup language would need to include the capability to define:

- Reference parameters (wing area, chord, span, and moment reference center),

- Physical properties (vehicle shape, mass properties, pilot eyepoint location) and collision volume or control points

- Documentation and provenance of data

- Dynamic model equations (in programming language independent format) and function data

Confidence bounds for data  
Specification for handling extrapolation at boundaries of data tables  
Static trim data (for validation)  
Dynamic maneuver time history data (for validation)

#### Standard to codify common modeling conventions

Although not included in the XML-based exchange documentation, a set of conventions would be outlined in the specification for unambiguous interpretation of the model data. These conventions would address, for example,

Use of 1992 ANSI/AIAA Recommended Practice for axes, symbology, and sign conventions  
Data dictionary, allowing deviation from the 1998 AIAA Flight Sim TC variable names  
Specify sets of units and their abbreviations  
Specify nominal equations of motion including earth / geodesy model  
Specify highest-level modularity: inertia, aerodynamics, propulsion, landing gear, and flight control models including inputs and outputs

#### Sample application: Aerodynamic Model for F-16

A popular, non-trivial aircraft model is given in Stevens & Lewis<sup>7</sup> for an F-16 fighter aerodynamic model. This has been refined and implemented as MATLAB-based software by Garza & Morelli<sup>8</sup>. This model has been realized in XML using a draft DTD developed by the authors. An excerpt appears in listing 1 with just one axis defined; a complete aero model representation in XML (and a draft DTD) has been developed and is being evaluated.

#### Next Steps

The first order of business would be the establishment, with support from industry and government, of a small group reporting to the Flight Simulation Technical Committee, which would solicit and evaluate proposed XML-based format specifications.

After a draft DAVE specification has been proposed, a pilot project involving two or more flight simulation facilities should be arranged to demonstrate a rapid transfer of an example flight dynamic model between facilities. Lessons learned from this exercise would be used to refine the specification.

Ultimately, applications could be written for a variety of platforms that natively understand the DAVE standard and would allow useful analysis and manipulation of a DAVE-compliant model. These could range from pseudo-real-time simulations on

desktop machines to aero database development applications on UNIX computers.

#### Concluding Remarks

XML is a candidate format for exchanging flight dynamic models while adhering to previous AIAA modeling data standards. Benefits include more efficient exchange of models between facilities which could lead to extremely rapid re-hosting efforts, and substantial cost savings. The gaming community is leading this effort with simple model descriptions in XML-formatted text files. The AIAA Modeling and Simulations Technical Committee should consider implementing an XML-based schema for flight dynamic model exchanges.

#### References

<sup>1</sup> World-Wide Web Consortium (W3C) recommendation: Extensible Markup Language (XML) 1.0 (Second Edition), 6 October 2000.

<sup>2</sup> Berndt, Jon S.: JSBSim model file format, as incorporated in FlightGear open-source Linux simulation project:

<http://jsbsim.sourceforge.net/ConfigurationFiles.pdf>

<sup>3</sup> Jackson, E. Bruce: Results of a Flight Simulation Software Methods Survey, AIAA 95-3414. Presented at the AIAA Flight Simulation Technologies Conference, Baltimore MD, August 1995.

<sup>4</sup> American Institute of Aeronautics and Astronautics, sponsor: Recommended Practice: Atmospheric and Space Flight Vehicle Coordinate Systems, ANSI/AIAA R-004-1992, February 1992.

<sup>5</sup> Hildreth, B. L.: A Process for the Development of Simulation Standards, AIAA Paper 94-3430, Presented at the AIAA Flight Simulation Technologies Conference, August 1994.

<sup>6</sup> Hildreth, Bruce L.: The Draft AIAA Flight Mechanics Modeling Standard: An Opportunity for Industry Feedback. AIAA 98-4576. Presented at the AIAA Flight Simulation Technologies Conference, Boston, MA, August 1998.

<sup>7</sup> Stevens, Brian L. and Lewis, Frank L.: Aircraft Control and Simulation. John Wiley & Sons, Inc., 1992. ISBN 0-471-61397-5.

<sup>8</sup> Garza, F. R.; and Morelli, E. A.: A Collection of Nonlinear Aircraft Simulations in MATLAB. NASA Technical Memorandum (publication pending).

### **Listing 1. Annotated excerpt from F-16 aerodynamic model XML file**

This is an excerpt of the Stevens & Lewis F-16 aerodynamics model, as modified by Garza & Morelli, realized in a candidate XML application for aerodynamic models. Included are annotations in Times font, with the actual XML shown in Courier font. This XML specification would apply to high-fidelity engineering simulation models; a simpler variant has been devised for lower-fidelity models containing much less information and details.

XML is quite similar to HTML format. Content (actual useful data) is delimited by tags (found in angle brackets) which provide the context for the data. Generally XML tags appear in pairs: an opening *<element>* tag followed by a closing *</element>* tag surrounding the content.

Tags can contain attributes: fields within the opening tag associated with a value: *<element attribute1="value">*. The value must be contained in single or double quotes.

The order of the tags is important and is specified in the referenced Data Type Definition (DTD) file (not shown). The DTD also specifies what attributes are allowed and/or required.

The content contained in XML tags may contain any UNICODE character (an extension of ASCII characters) except the opening angle bracket ("*<*") character.

The excerpt below captures only a portion of the F-16 model, generally showing only one of each type of element. The file starts with header information that identifies it as an XML file that follows a content model outlined in a separate DTD file which defines the Dynamic Aerospace Vehicle Exchange (DAVE) XML element set (not shown).

<code>&lt;?xml version="1.0" standalone="no"?&gt;</code>	Required first line for XML files
<code>&lt;!DOCTYPE DAVE SYSTEM "DAVE.dtd"&gt;</code>	Specifies the Data Type Definition (schema)
<code>&lt;DAVE&gt;</code>	Opening tag identifies top-level element as exchange doc
<code>&lt;fileHeader&gt;</code>	Subsidiary tag identifies header material with author, date, etc.
<code>&lt;author name="Bruce Jackson" org="NASA Langley Research Center" xns="@bjax"/&gt;</code>	
<code>&lt;fileCreationDate date="28-MAR-2002"/&gt;</code>	
<code>&lt;description&gt;</code>	Required section to provide explanation of contents
F-16 Aero Data file. Based on Morelli's adaptation of	
Stevens and Lewis' F-16 example [1] described in Garza &	
Morelli's TM [2]. Obtained from E. A. Morelli in the form of	
Matlab scripts [3].	
<code>&lt;/description&gt;</code>	

Following general information about this model is a set of referenced documents that should be used to describe the origins of the data.

<code>&lt;!-- ===== --&gt;</code>	Comments in XML are delimited by <code>&lt;!--</code> and <code>--&gt;</code> strings
<code>&lt;!-- References --&gt;</code>	
<code>&lt;!-- ===== --&gt;</code>	This section establishes refIDs that are referred to later

Note the use of the empty `<reference/>` tag with attribute values providing the desired information.

<code>&lt;reference refID="REF01" author="Stevens, Brian L. and Lewis, Frank L." title="Aircraft Control and Simulation" accession="ISBN 0-471-61397-5" date="1992"/&gt;</code>	
<code>&lt;reference refID="REF02" author="Garza, F. R.; and Morelli, E. A." title="A Collection of Nonlinear Aircraft Simulations in MATLAB" accession="NASA TM-2002-xxxxxx" date="JUN-2002"/&gt;</code>	
<code>&lt;reference refID="REF03" author="Morelli, Eugene A." title="f16_aero_setup.m" date="17-JUN-1995"/&gt;</code>	
<code>&lt;/fileHeader&gt;</code>	Matches <code>&lt;fileheader&gt;</code> above to denote end of section



What follows next are variable definitions. These roughly correspond to programming language variables, such as input, output, and intermediate (local) variables, but include a machine-language identifier (the varID), a human-readable name, the units of measure expected, description, and a Greek symbol defined as a UNICODE hexadecimal value. The variable definition can also include a mathematical expression specifying how the variable is calculated based on other variables - in essence, the build-up equations. These general mathematical expressions are built from MathML syntax - another XML application gaining wide acceptance.

```

<!-- ===== -->           Comments contained within special character sequences
<!-- Input variables      -->
<!-- ===== -->

<variableDef name="alpha" varID="alpha" units="deg" symbol="#x3B1">
  <description>
    Instantaneous true angle-of-attack, in degrees
  </description>
</variableDef>
<variableDef name="beta" varID="beta" units="deg" symbol="#x3B2">
  <description>
    Instantaneous true angle-of-sideslip, in degrees. Positive wind-in-the-right-ear.
  </description>
</variableDef>
.
.
.
<!-- ===== -->
<!-- Local variables      -->
<!-- ===== -->

<variableDef name="absbeta" varID="absbeta" units="deg">
  <description>
    Absolute value of angle-of-sideslip, deg.
  </description>
  <calculation>
    This variable is the absolute value of another variable, beta.
    <math>
      <apply><abs/><ci>beta</ci></apply>
    </math>
  </calculation>
</variableDef>
.
.
.
<!-- ===== -->
<!-- Output variables     -->
<!-- ===== -->

<variableDef name="CZ0" varID="CZ0" units="ND" sign="+down">
  <description>
    Basic coefficient of force in the Z-body direction (+DWN)
  </description>
</variableDef>
.
.
.

```

**Listing 1 (cont'd). Annotated excerpt from F-16 aerodynamic model XML file**

The next required element are breakpoint values. These are kept separate from the function definitions since a particular breakpoint set may be used by more than one function. These are defined by their unique breakpoint ID (bpID) attribute.

```
<!-- ===== -->
<!-- Breakpoint values -->
<!-- ===== -->

<breakpointDef name="alpha" bpID="ALPHA1" units="deg">
  <description>
    Alpha breakpoints for basic and damping aero tables
  </description>
  <bpVals>
    -10., -5., 0., 10., 15., 20., 25., 30., 35., 40., 45.
  </bpVals>
</breakpointDef>
.
.
.
```

The function tables are described by <function> elements that refer to source documents (i.e. wind tunnel test reports). The input variables are identified in sequence, along with instructions for dealing with out-of-bound inputs. The table data itself is specified, along with references to the appropriate breakpoint sets.

```
<!-- ===== -->
<!-- Basic Functions -->
<!-- ===== -->

<function name="Basic CZ">
  <description>
    Basic coefficient of Z-force table as a function of angle of attack
  </description>
  <provenance>
    <author name="Bruce Jackson" org="NASA Langley Research Center" xns="@bjax"/>
    <functionCreationDate date="28-MAR-2002"/>
    <documentRef docID="REF01"/>
    <documentRef docID="REF02"/>
    <documentRef docID="REF03"/>
  </provenance>
  <independentVarRef varID="alpha" min="-10.0" max="45.0" extrapolate="neither"/>
  <dependentVarRef varID="CZ0"/>
  <functionDefn name="CZ0_fn">
    <griddedTable name="CZ0_table">
      <breakpointRefs>
        <bpRef bpID="ALPHA1"/>
      </breakpointRefs>
      <dataTable>
        .770,.241,-.100,-.416,-.731,-1.053, -1.366,-1.646,-1.917,-2.120,-2.248,-2.229
      </dataTable>
    </griddedTable>
  </functionDefn>
</function>
.
.
.
</DAVE>
```

Matches opening <DAVE> tag at top of file

**Listing 1 (concluded). Annotated excerpt from F-16 aerodynamic model XML file**