Introduction to DAVE-ML

Flight Dynamic Model Exchange using XML

December 3, 2002
Bruce Jackson
<e.b.jackson@larc.nasa.gov>

What is XML?

- XML (like HTML) is based on SGML, an early ISO standard for electronic information storage & exchange
- XML is a flexible, extensible meta-markup language where information is delineated by contextual tags
- XML is not a programming language

XML syntax

- Elements (info surrounded by tag pair)

```
<mytag>information</mytag>
```

- Attributes (info embedded in tag)

```
<mytag important="yes">information</mytag>
```

- Empty elements (single tag as flag)

```
<useEnglishUnits/>
```

- Comments

```
<!-- a comment goes here -->
```

Required header

- Each XML document should start with a single-line xml declaration:

```
<?xml version-"1.0" encoding="US-ASCII" standalone="yes"?>
```

- This identifies it as XML without a external grammar specification

Validation

- XML documents must be *well-formed* (meaning all start- and end-tags are paired and correctly nested, etc.)
- If using an external grammar, an XML document is *valid* if it respects the gammar rules in the external document
- Special XML editors can determine validation (emacs, BBEdit, others)

Document Type Definition (DTD)

- A DTD is an XML document that defines a particular grammar - that is, the allowed elements and attributes and their relationship to each other
- Math-ML, for example, has a DTD that defines allowable mathematical elements
- DAVE-ML consists of an evolving DTD

Math-ML

- Math-ML is a W3C-adopted XML-based grammar for mathematical information
- Includes specifications for both notation (how the equation appears) and content (how the relationship works)
- Should allow useful, interactive, hi-quality mathematical expressions on Web pages
- Supported by Wolfram Research (Mathematica)

AIAA standards

- DAVE-ML proposed to implement draft AIAA standard practice regarding
  - Variable names
  - Function table information
- Additional standards may be rolled in
- DAVE-ML may be offered to AIAA as standard

DAVE-ML grammar

- Intended to encode complete high-fidelity flight model in non-proprietary, facility-independent way
- First step: static models (tables & equations)
  - Aerodynamic model
  - Inertial subsystem models
- Next step:
  - Checkcase data (static & dynamic)
  - Time-history data
- Eventually include dynamic elements

DAVE-ML top-level elements

- fileHeader - Contains author, creation date, description, references, modification record
- variableDef(s)
  - Describes inputs, outputs, intermediate signals
  - Includes build-up equations with basic math functions
  - Equations can reference table outputs
- breakpointDef(s) - Used to store breakpoint sets which may be reused
- function(s) - Marries breakpoint sets with independent variables to form output (dependent) variable

fileHeader element details

- author
- fileCreationDate
- description (optional)
- reference(s) (optional)
- modificationRecord(s) (optional)

fileHeader example

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DAVEfunc SYSTEM "DAVEfunc.dtd">
<DAVEfunc>
    <fileHeader>

        <author name="Bruce Jackson" org="NASA Langley Research Center" xns="@bjax"/>
        <fileCreationDate date="28-MAR-2002"/>
        <description>
      F-16 Aero Data file. Based on Morelli's adaptation of
          Stevens and Lewis' F-16 example [1] described in Garza &amp;
          Morelli's TM [2]. Obtained from E. A. Morelli in the form of
          Matlab scripts [3] &amp; [4]
    </description>

      <!-- ================== -->
      <!--     References     -->
      <!-- ================== -->

      <reference refID="REF01" author="Stevens, Brian L. and Lewis, Frank L."
          title="Aircraft Control and Simulation"
          accession="ISBN 0-471-61397-5" date="1992"/>

      <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"/>

      <reference refID="REF03" author="Morelli, Eugene A."
          title="f16_aero.m" date="17-JUN-1995"/>

      <reference refID="REF04" author="Morelli, Eugene A."
          title="f16_aero_setup.m" date="17-JUN-1995"/>

    </fileHeader>
```

`variableDef` element details

- Attributes used to define name, assign unique variable ID, specify units
- Optional attributes include axis system, sign, alias, and symbol (UNICODE)
- Sub-elements
  - `description` (optional)
  - `calculation` (optional - uses mathML 2.0)
  - Empty `<isOutput/>` element signifies intermediate output quantity

variableDef examples (constants)

```
<variableDef name="rtd" varID="rtd" units="deg/rad">
   <description>
   Conversion constant from radians to degrees
   </description>
   <calculation>
     <math>
   <apply>
   <quotient/>
     <cn>180.</cn>
     <cn>3.14159265</cn>
   </apply>
     </math>
   </calculation>
 </variableDef>

 <variableDef name="xcgr" varID="xcgr" units="fract" initialValue="0.35">
   <description>
   Default location of center of gravity relative to wing leading
   edge, expressed as a fraction of aerodynamic chord length.
   </description>
 </variableDef>
```

variableDef examples (inputs)

```
<variableDef name ="True_Airspeed_f_p_s" varID="vt" units= "ft/sec">
   <description>
    True airspeed, ft/sec
   </description>
 </variableDef>

 <variableDef name="Angle_of_Attack_deg" varID="alpha" units="deg" symbol="#x3B1">
   <description>
      Instantaneous true angle-of-attack, in degrees
   </description>
 </variableDef>

 <variableDef name=" Angle_of_Sideslip_deg " varID="beta" units="deg" symbol="#x3B2"
    sign="+wind in right ear">
   <description>
      Instantaneous true angle-of-sideslip, in degrees
   </description>
 </variableDef>
```

variableDef examples (intermediate calculations)

```
<variableDef name="drdr" varID="drdr" units="">
   <description>
   Normalized rudder deflection.
   </description>
   <calculation>
     <math>
   <apply>
     <quotient/>
     <ci>rdr</ci>
     <cn>30.0</cn>
   </apply>
     </math>
   </calculation>
 </variableDef>
```

variableDef example (output calculation)

```
<variableDef name="Cl0" varID="clt" units="" sign="+right wing down">
   <description>
     Basic coefficient of moment around the X-body direction (roll) (+RWD)
   </description>
   <calculation>
     <math>
       <apply>
      <piecewise>
        <piece>
          <apply><minus/><ci>absCl0</ci></apply>
          <apply><lt/><ci>beta</ci><cn>0</cn></apply>
        </piece>
        <otherwise>
          <ci>absCl0</ci>
        </otherwise>
      </piecewise>
    </apply>
     </math>
   </calculation>
 </variableDef>
```

breakpointDef element details

- Attributes define unique bpID, optional name and units
- Sub-elements
  - description (optional)
  - bpVals

breakpointDef examples

```
<breakpointDef name="beta" bpID="BETA1" units="deg">
   <description>
      Angle-of-sideslip breakpoints for basic aero tables
   </description>
   <bpVals>
       0.0, 5.0, 10.0, 15.0, 20.0, 25.0, 30.0
   </bpVals>
 </breakpointDef>

 <breakpointDef name="beta" bpID="BETA2" units="deg">
   <description>
      Angle-of-sideslip breakpoints for control power tables
   </description>
   <bpVals>
       -30.0, -20.0, -10.0, 0.0, 10.0, 20.0, 30.0
   </bpVals>
 </breakpointDef>
```

`function` elements

- Sole attribute: `name`
- Sub-elements
  - `description` (optional)
  - `provenance` (optional)
  - Two styles of functions:
    - Simple 1-D (`independentVarPts`, `dependentVarPts`)
    - n-D `functionDefn` with `varRefs`

n-D functions

- Three elements:
  - – independentVarRef(s) using varID
    - Optional min, max, extrapolate attributes
    - Listed order is important
  - – dependentVarRef using varID
  - – functionDefn (with optional name attribute)
    - Two styles of tables
      - – Gridded table (orthogonal)
      - – Ungridded table (point-by-point)

griddedTable element

- Sole attribute: name (optional)
- Sub-elements
  - breakpointRefs - list of bpRef(s)
  - Optional confidenceBound value
    - Multiplicative 3-sigma ± variation
  - dataTable
    - Comma-separated list of gridded table values in which the last bpRef changes most rapidly

functionDefn example

```
<function name="Basic Cl">
   <description>
     Basic coefficient of rolling moment as a function of angle of attack and sideslip angle
   </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="absbeta" min="0.0"   max="30.0" extrapolate="neither"/> <!-- Beta breakpoints -->
   <independentVarRef varID="alpha"   min="-10.0" max="45.0" extrapolate="neither"/> <!-- Alpha breakpoints -->
   <dependentVarRef varID="absCl0"/>

   <functionDefn name="Cl0_fn">
     <griddedTable name="Cl0_table">
    <breakpointRefs>
      <bpRef bpID="BETA1"/>
      <bpRef bpID="ALPHA1"/>
    </breakpointRefs>
    <dataTable> <!-- Note: last breakpoint changes most rapidly -->
   0.,   0.,   0.,   0.,   0.,   0.,    0.,   0.,   0.,   0.,   0.,   0.,
-.001,-.004,-.008,-.012,-.016,-.022, -.022,-.021,-.015,-.008,-.013,-.015,
-.003,-.009,-.017,-.024,-.030,-.041, -.045,-.040,-.016,-.002,-.010,-.019,
-.001,-.010,-.020,-.030,-.039,-.054, -.057,-.054,-.023,-.006,-.014,-.027,
 .000,-.010,-.022,-.034,-.047,-.060, -.069,-.067,-.033,-.036,-.035,-.035,
 .007,-.010,-.023,-.034,-.049,-.063, -.081,-.079,-.060,-.058,-.062,-.059,
 .009,-.011,-.023,-.037,-.050,-.068, -.089,-.088,-.091,-.076,-.077,-.076
       </dataTable>
     </griddedTable>
   </functionDefn>
 </function>
```

ungriddedTable element

- Sole attribute: name (optional)
- Sub-elements
  - Optional confidenceBound value
    - Multiplicative 3-sigma ± variation
  - dataPoint(s), each contains a comma-separated list:
    - Independent variable value (in order)
    - Dependent value
    - –3-sigma value (optional)
    - +3-sigma value (optional)
    - refID (optional)

What's next?

- Demo of exchange of aero models
- Feedback to refine grammar
- Develop tools - editors, report generator
- Establish informal working group for refinement
- Work towards AIAA standard
- Establish test cases for new facilities