The Proposed AIAA Standard for Simulation Model Exchange

American Institute of Aeronautics and Astronautics

From Science to Solutions™
SAIC.COM

Standards Promote Productivity

- Improved information exchange
  - More accurate simulations
  - More consistent simulations
  - Lower cost
- Improved interoperability
- Proper s/w reuse

Existing standards

- Simnet/DIS/HLA-networking/architecture
- SEDRIS- environmental data representation
- FAA Advisory Circulars (AC 120-40)
- Standard atmosphere
- Standard world (WGS –95?)

Why haven't we done better?– we should be embarassed

- Afraid of competition (proprietary)
- Standards are a long term investment
  - Up front cost
  - Hard to document return
- Cultural barriers
  - "Pet" methods (not necessarily even correct)
  - Reuse aversion
  - Simulation "club"

Why we should do better

- Responsibility- we have a commitment to our user community, we shouldn't waste money, we should use the money to simulate better
- Longevity- if you want to exist in 20 years you need to spend some effort in long term investments
- Productivity- (same as longevity)- if you don't do it better you'll be left behind

History of Vehicle Dynamic Standards

- M&S T.C. started standards effort in early 1990’s
- Efforts focused on vehicle dynamics
- Objective: to facilitate the exchange of a math model from one site to another
- Current status:
  - Standard developed (Mod 1)
  - XML tested as an method of implementation

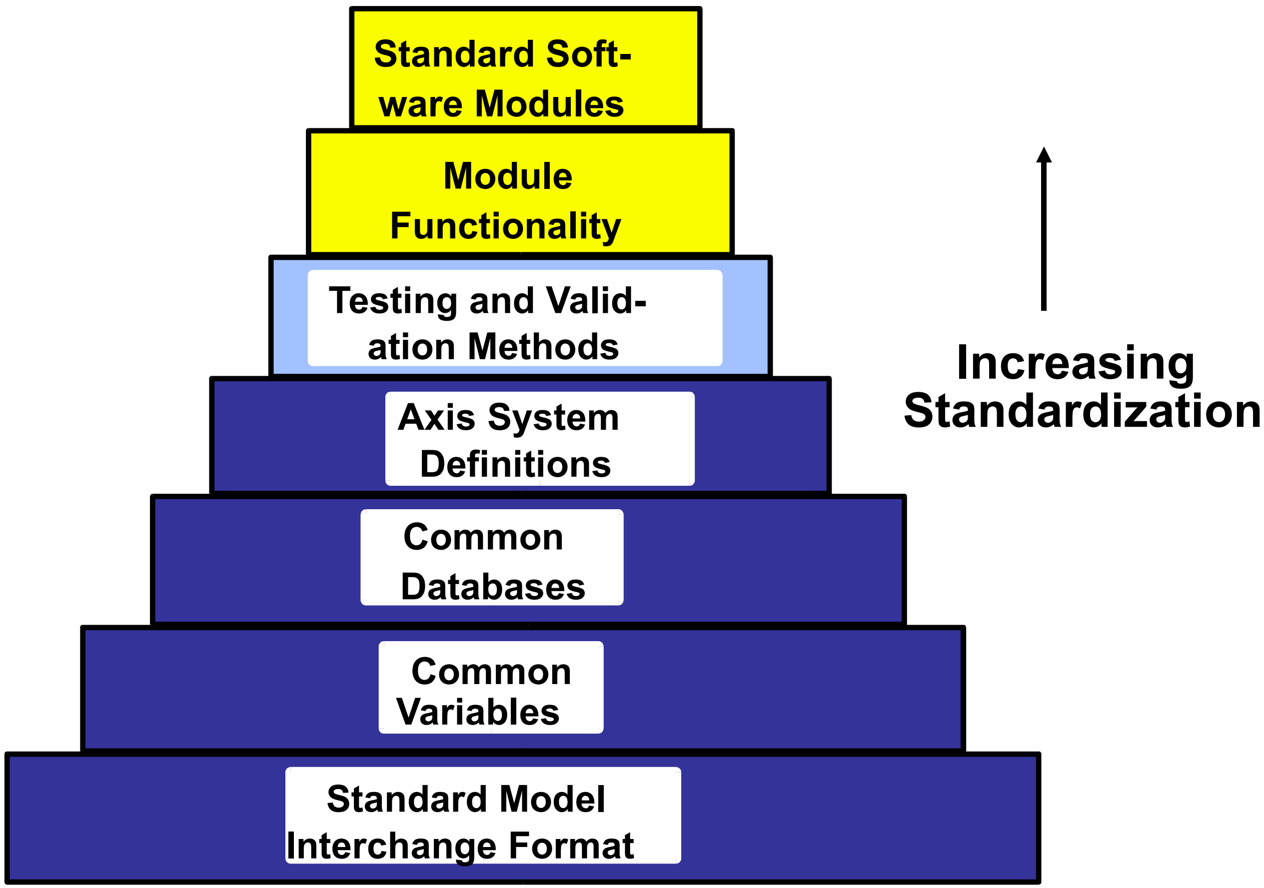
Concept

- Need for standard representation of vehicle dynamics/aerodynamics
- Get away from ad-hoc, site-specific “standards”
- Many are possible- we’re proposing one
- Standard is superset of typical site-specific standards
- “Visual database-like” import/export from/to standard
- No requirement for internal use in your simulator!

Business Case Summary

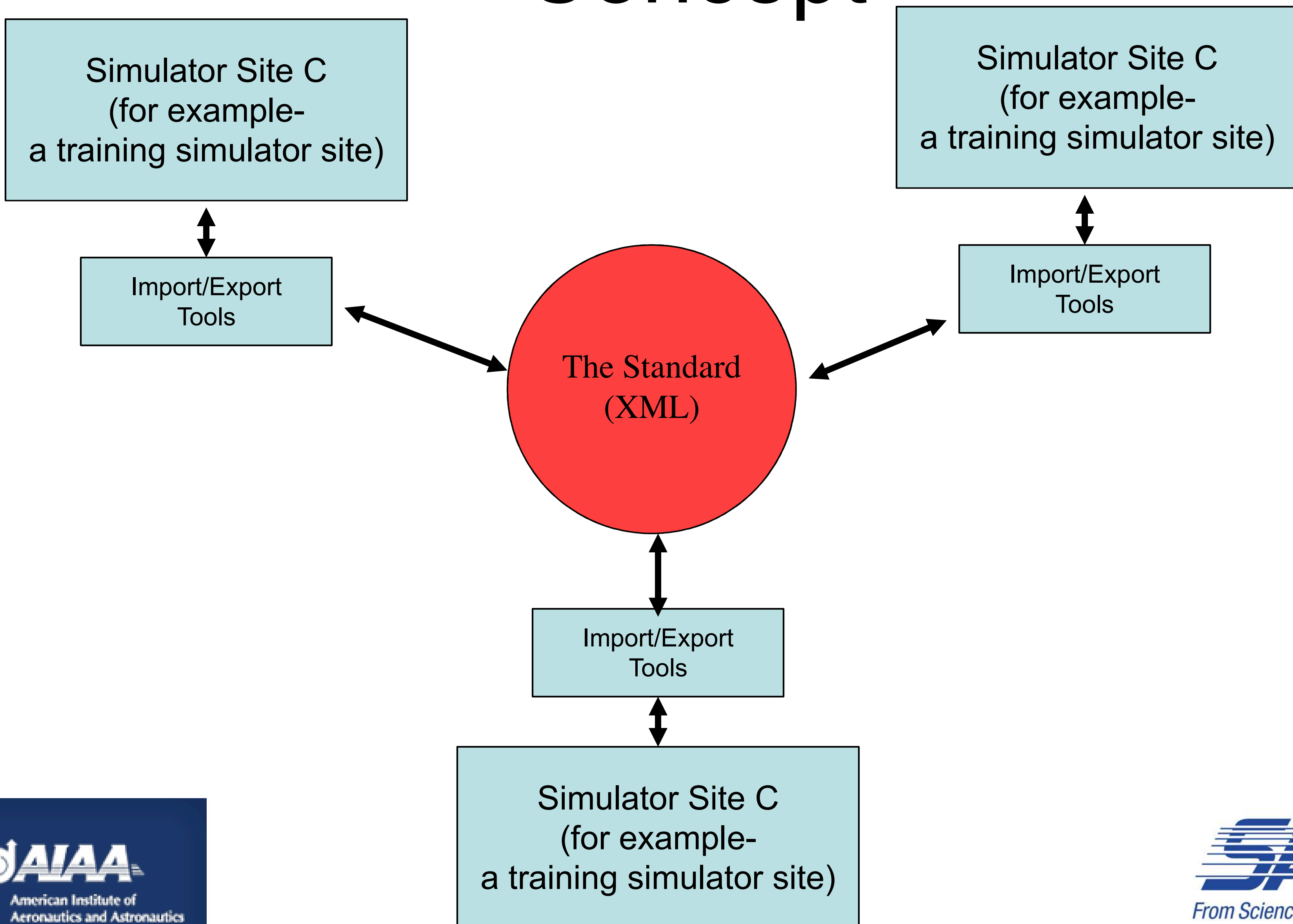
- Conservative analysis: $6.8M+ savings/yr.
- Typical case for a military aircraft
- Results in an average savings of $117K per year per simulator
- Savings only makes sense when applied to the whole community (this type of a/c)
- Savings to the entire simulator industry is many times this amount

Simulation Standards Development Road Map



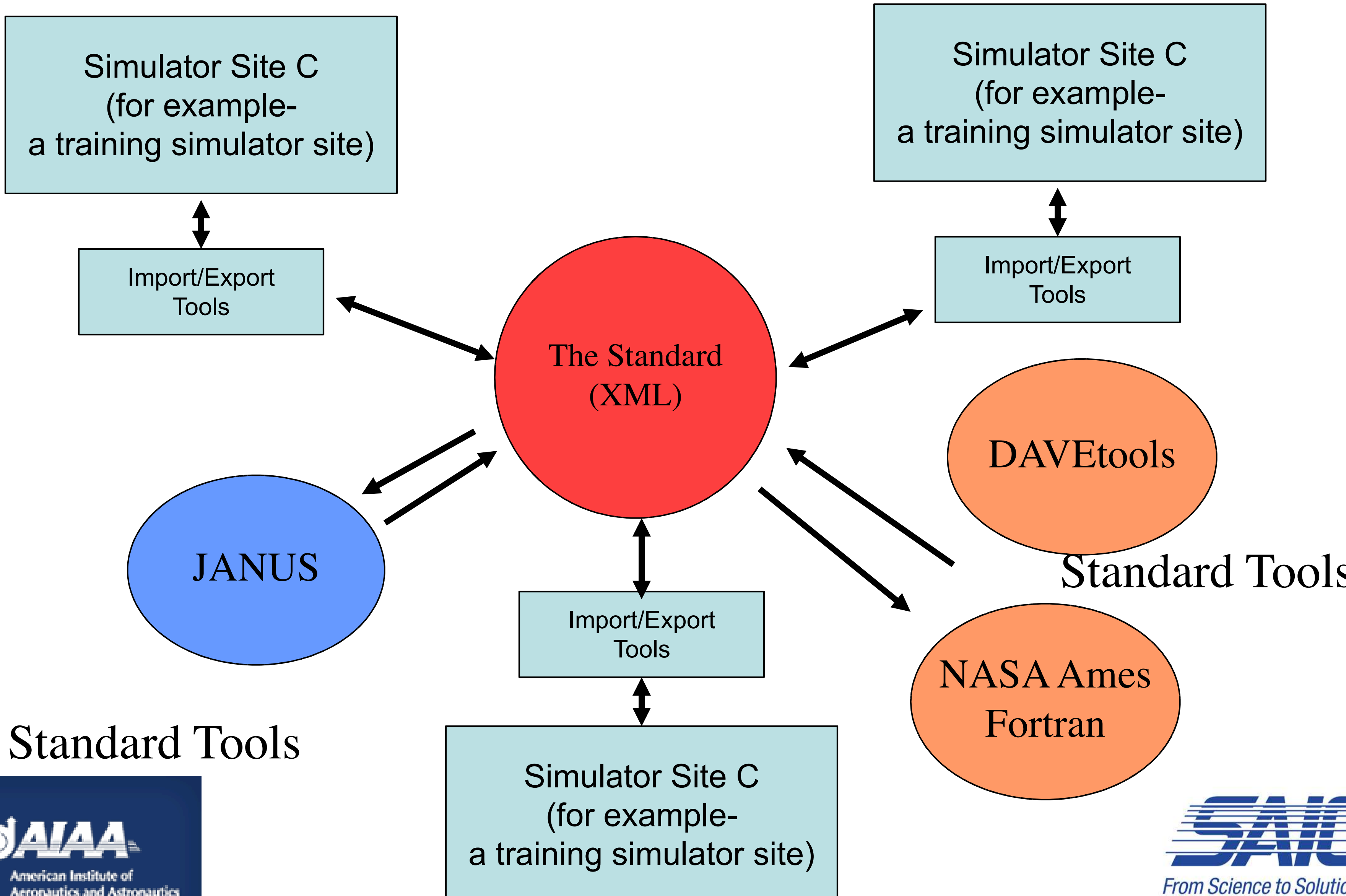
Each Level Builds Upon the Other

Concept



Simulator Site C
(for example-
a training simulator site)
Import/Export
Tools
Simulator Site C
(for example-
a training simulator site)
Import/Export
Tools
The Standard
(XML)
Import/Export
Tools
Simulator Site C
(for example-
a training simulator site)

Tools to support the standard

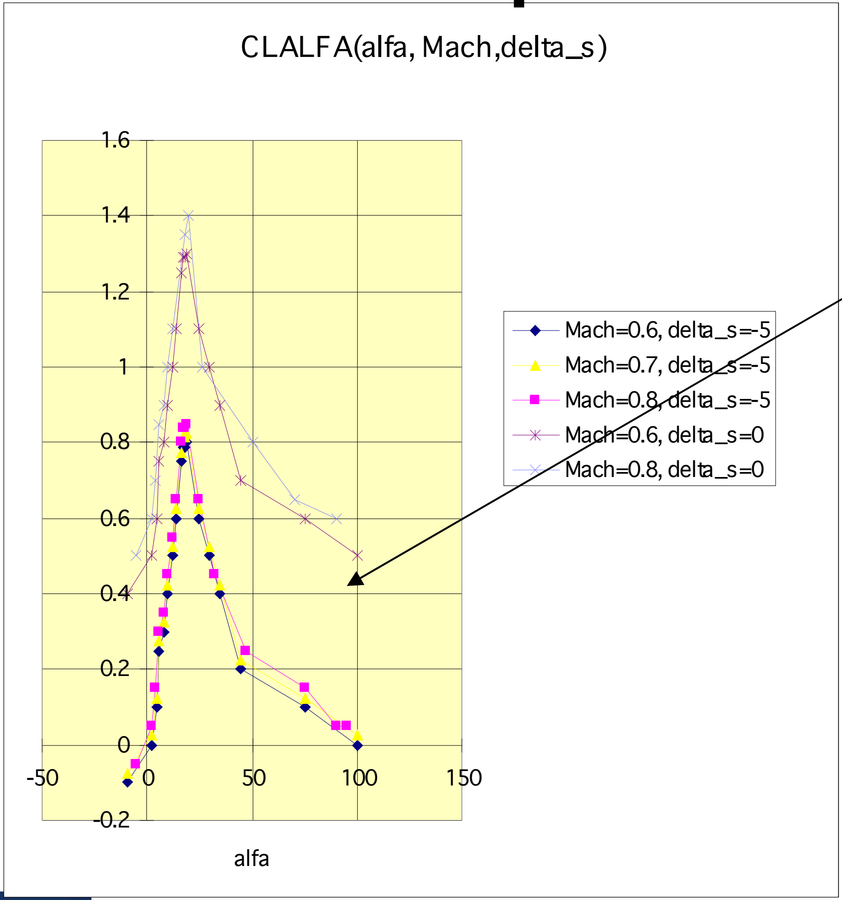


4 Key Requirements for a Standard

1. Function table data- required to transfer non-linear model components-standard adds:
   - Provenance
   - Statistics
2. Time history data- required to verify proper model transfer
3. Axis system definitions
4. Definitions (variable names)- required to clearly state what the transferred information is (units, axis system, sign convention, etc)

1. Function Table Data

Function table-with statistics and provenance



Four elements per data point

Independent variables

Reference

s, Mach, , C_L, ref, - ,

0.0, 0.8, 90, 0.60, C, -.0032, 0068

Dependent variable

Statistics

The statistics data (the confidence intervals) are optional

Applicable to automatic Monte Carlo studies

2. Time History Data

Time History Data

- Required for model verification-any model exchange should include simulation time histories to allow model verification
- Use of standard variable names (optional) and axis systems (optional) helps clearly define the validation data
- Simulation time histories are a subset of flight test data
- Allows the simulation community to leverage the flight test data I/O APIs.

3. Axis Systems

American Institute of Aeronautics and Astronautics

From Science to Solutions™
SAIC.COM

Axis Systems

- Use the overlap of existing AIAA/ANSI Recommended Practice R-004-1992 and DIS 3.5 Axis Systems
  - Body axis system
  - Earth fixed axis system
- Addition of a Flat Earth (local) axis system for convenience

4. Variable Names

Proposed AIAA Standard-Definitions (Variable Names)

- Standard Library or "Datapool" of variable names
- Objective:
  - Clear definition of the significant components and parameters of a model and its validation data.
  - For example:
    - Angle of attack—has many similar but SIGNIFICANTLY different meanings
      - wing angle of attack
      - fuselage angle of attack
      - angle of attack with/without turbulence effects
      - in degrees or radians
      - ranging from ± 90 or ± 180 degrees
  - Extremely important in validation.

American Institute of Aeronautics and Astronautics

Clearly Defined Variable Names Critical to Communication

From Science to Solutions™

SAIC.COM

Variable Names – KEY POINTS

- Variable Naming convention includes:
  - Identification of Simulation States and Inputs
  - Units- either English or SI
- Linear System Formulation
  - x = states
  - u = inputs (or controls)

$$dx/dt = Ax + Bu$$
$$Y = Cx + Du$$

States and Inputs are Key – Everything in the dynamic simulation depends upon them

They should be easily identifiable for good software documentation and maintainability

Units for clarity and documentation purposes

Variable Naming Convention

- Each name has up to six components
  - (prefix) (variable source domain) (axis or reference system) (specific axis or reference) (core name) (units)
  - Similar to C naming convention
- Examples
  - s_bodyXVelocity_fps — s_ prefix indicates that this variable is a state
  - sd_bodyXAcceleration_fps2 — sd_ prefix indicates that this variable is a state derivative
  - aeroXBodyForceCoefficient
  - thrustYBodyForce_lbf

The point of standard variable names is simply to help clearly define the information being exchanged

Variable Names – Issues – Units

- Why units? Compare

  CLFlaps0 = CLAlfa*angleOfAttack + CLDe * De + CLQ*QB*chord/(2.0*trueAirspeed)

  vs

  CLFlaps0 = CLALFA_prad*qngleOfAttack_rad + CLDe_pdeg*De_deg+ CLQ*s_bodyPitchRate_radps*chord_f/(2.0*TrueAirspeed_fps)

  vs

  CLFlaps0 = CLALFA_pdeg*qngleOfAttack_deg + CLDe_pdeg*De_deg+ CLQ*s_bodyPitchRate_degps*chord_f/(2.0*TrueAirspeed_fps)

Example Variable Names

- s_BodyXVelocity_fps
- sd_BodyXAcceleration_fps2
- GEAxisZVelocity_fps
- s_BodyRollRate_radps
- YBodyThrustForce_lbf

Variable Names – Units

- Conclusion – Units included makes code
  - More self documenting
  - Less ambiguous
  - Works for English or Metric System
  - Helps catch homogeneity of units errors
  - Longer to type (However typing is by far the shortest part of s/w development)

So what do the standard users work with?

DAVE-ML: The real utility to you!

- The standard has been realized in XML
- Tested in model exchanges between NAVAIR, Patuxent River, MD, and NASA Ames, Mountain View, CA.
  - Fortran, C and Simulink tools developed useful to all!
  - Demonstrated over an order of magnitude reduction in effort to export/import a model
  - Has matured the standard through use
  - Demonstrated the utility and flexibility of DAVE-ML

DAVE-ML will be the standard you use

American Institute of Aeronautics and Astronautics

From Science to Solutions™
SAIC.COM

Tools that facilitate standard implementation

DAVEtools (public domain)	Bruce Jackson NASA Langley Java package for manipulating DAVE-ML and Generation pf Simulink Models
JANUS (Not yet public domain)	DSTO (Australia) C API for manipulation of DAVE-ML models
NASA Ames FTP	Bill Cleveland Fortran tools for import/export to NASA Ames Format

Tools that facilitate standard implementation

NCSA HDF APIs (public domain)	APIs for input/output of HDF data
Lockheed Martin HDF APIs (readily available?)	APIs for input/output of HDF data. Tailored to the time history data format.

Time-History Data Standard-HDF 5

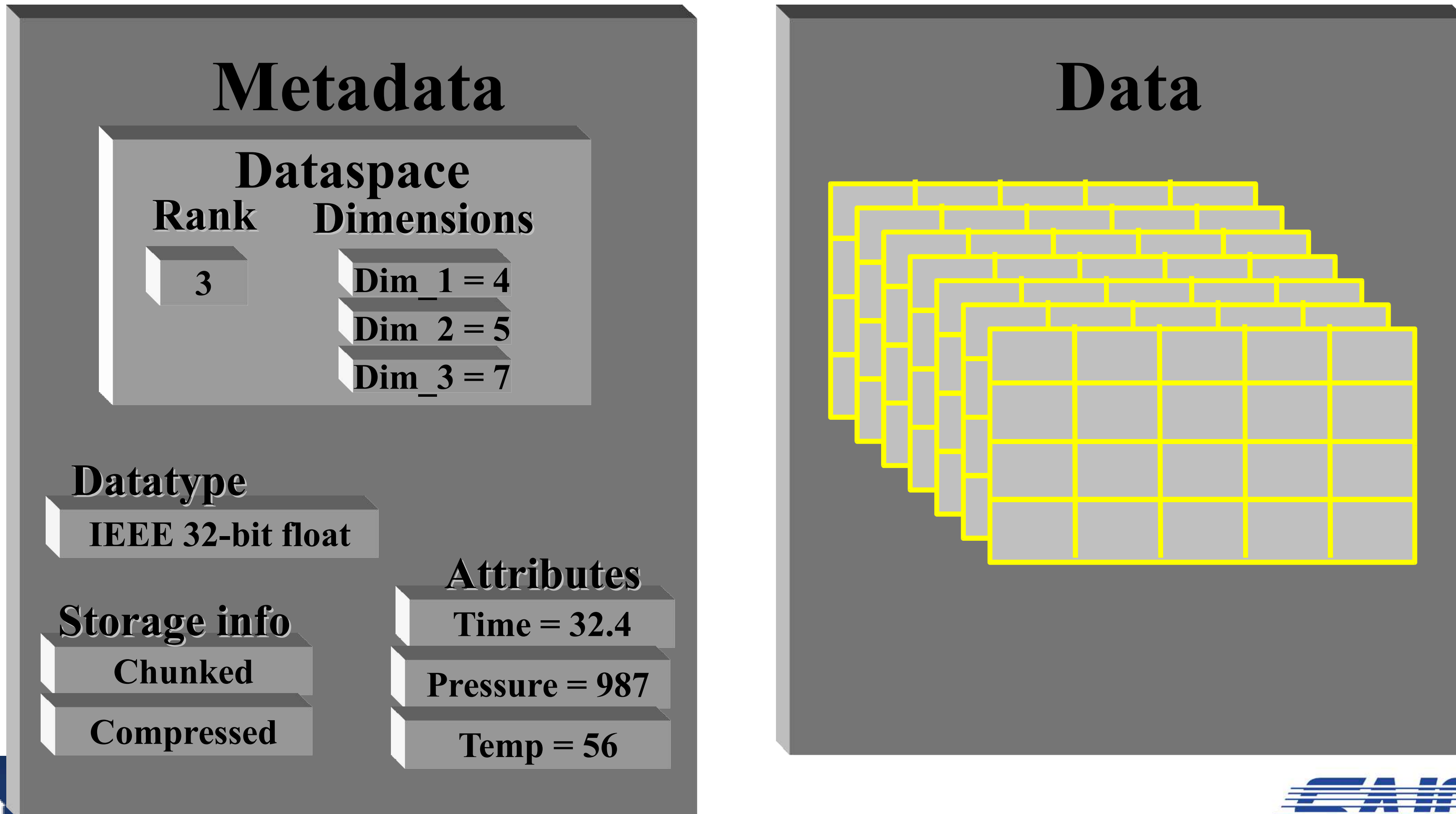
- JSF Flight test data standard
  - Mature-In use for JSF, F-16, C-5, C-130
  - Good compression
  - Works on virtually any platform
- HDF 5 format, publicly releasable
  - NCSA APIs publicly available
  - Lockheed Martin APIs may be releasable, are certainly available to some organizations
  - MATLAB has an HDF interface

No sense in reinventing a good wheel

What is HDF?

- Format and software for scientific data
- Stores images, multidimensional arrays, tables, etc.
- Emphasis on
  - Storage and I/O efficiency
  - Standards and platform portability
- Free and commercial software support
- Users from many engineering and scientific fields

HDF5 Datasets



HDF5 API Library

- High-Level Object API (C, Fortran 90, Java, C++)
  - Access objects (arrays, tables, images, packets)
  - Move and transform data
  - Combine many low-level API calls in common structure
- Low-Level API
  - Detailed access to all parts of HDF data
  - Several distinct interfaces

HDF5 Tools/Utilities

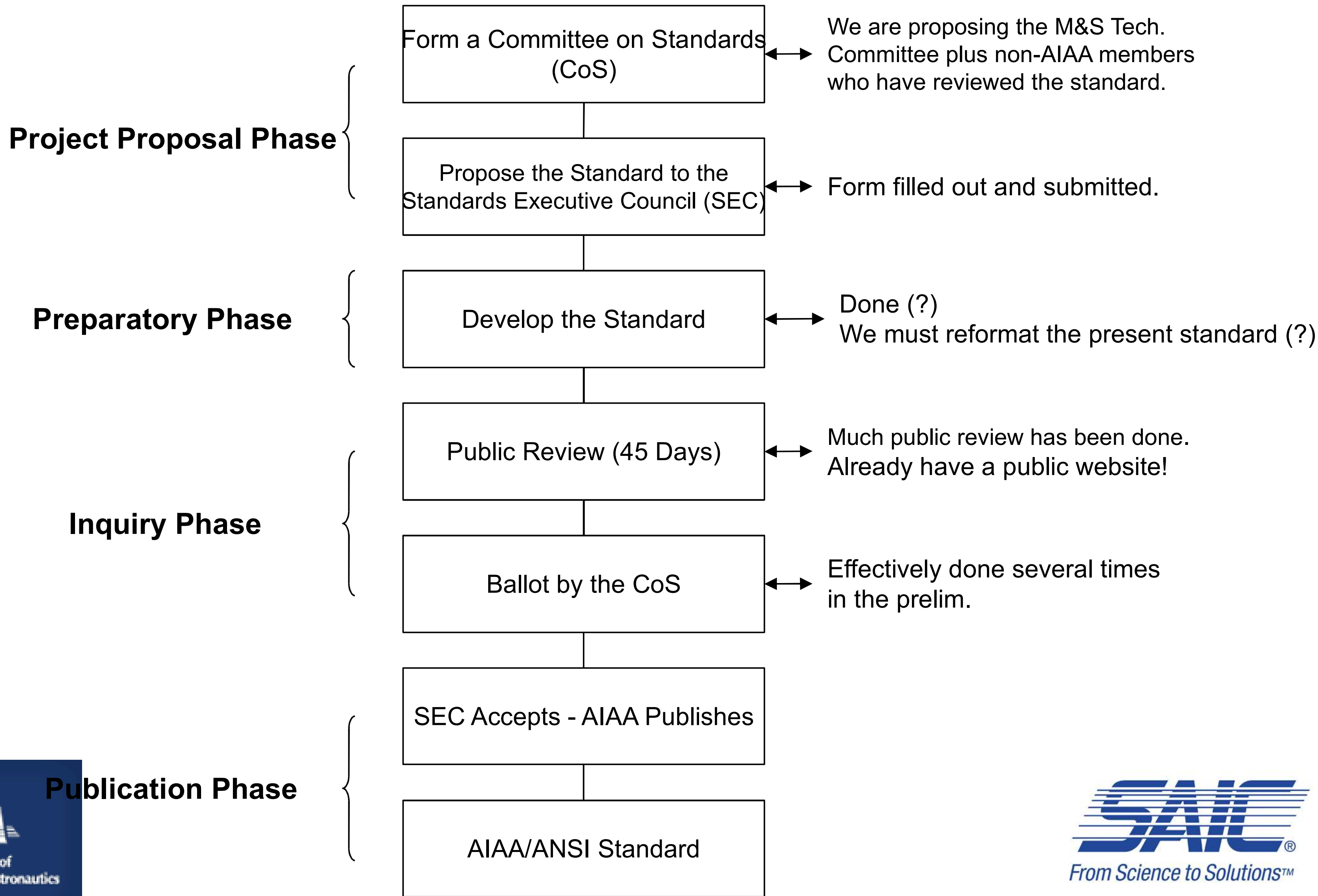
- Multiple tools provided by NCSA
  - Import and export from multiple formats
  - View content of HDF file
  - Partition file(s)
  - Conversion between HDF4 and HDF5
  - Java and Web-browser plugins
- Many commercial and freely available programs read/write HDF5 files
  - MATLAB, Mathematica, IDL

The Next Step- an AIAA/ANSI Standard

Submittal to the AIAA is in progress

The AIAA Standards Process

AIAA Process | **Progress Thus Far**



AIAA American Institute of Aeronautics and Astronautics

SAIC From Science to Solutions™ SAIC.COM

The Future- Maintaining, improving, and expanding

Life Cycle Support

- Any Standard must be supported and evolve over time to remain current
- User’s questions must be answered
  - A method of feedback must be maintained
  - Maintain web page
- Phone/E’Mail response?
- Annual updates?
- *Create/ maintain a catalog of models?*

Ongoing support is required for any standard

AIAA American Institute of Aeronautics and Astronautics

SAIC® From Science to Solutions™ SAIC.COM

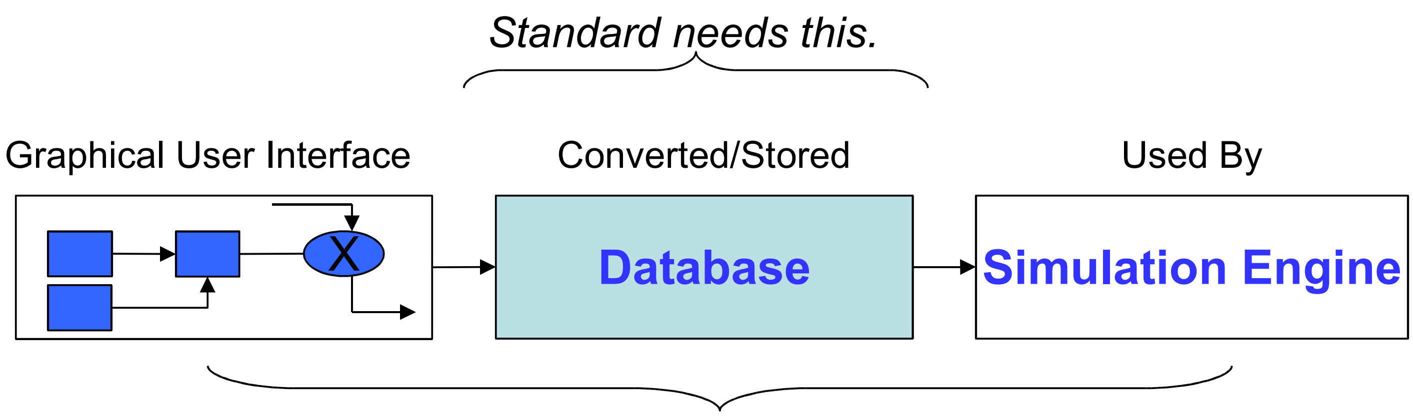
Addition of Controls and Dynamic Equations

- Standa can presently transfer algebraic equations
  - Typical aero and mass and inertia static equations
- Control systems and dynamic equations are the next challenge
  - Very brief discussions started with Mathworks

The Big Issue

- The graphical simulation database
  - Graphically defined and manipulated models are stored in a database
  - Conceptually the standard would be the definition of that database

Graphical Simulation Database



Mathworks has all this in Simulink.

Summary and Conclusions

Some benefits of the standard

- *Substantial savings of time & effort clearly demonstrated though use of the standard*
- *Verification of exchanged models clearly simplified by use of standard time history format and data definition*
- Function table Applicable to automatic Monte Carlo studies
- Easy to grow and change as technology requires

Feedback has been virtually universally positive

Summary

- Status of the standard- the standard is DAVE-ML and it is ready!
  - Ready and tested
    - Variable definitions
    - Axis systems
    - Simple math-DAVE-ML
    - Function data-DAVE-ML
    - Time history data-HDF 5
  - Application Programmer's Interfaces Available
  - Submittal to the AIAA is in progress

Summary

- Work will continue
  - Maintenance of the standard
  - Development and sharing of new/better APIs (by the community of use)
  - Development of dynamic equation capability
  - Development of control system data standards

Acknowledgements

The people who have made it happen

- Bruce Jackson- NASA Langley
- Tom Alderete- NASA Ames
- Bill McNamara- NAVAIR Patuxent River
- Brent York- NAVAIR Patuxent River (now INDRA)
- Bill Cleveland- Northrop Grumman/NASA Ames
- Geoff Brian and Hilary Keating-Ball Aerospace/DSTO support.

Back up

Conclusions

- The initial version of the standard is ready
  - Substantial savings of time & effort clearly possible
  - Improve efficiency of the simulation community
- DAVE-ML file definitions serve as complete model archive
  - Includes provenance, equations, data, statistics
  - Applicable to automatic Monte Carlo studies
  - Easy to grow and change as technology requires
- Exchange between NAVAIR and NASA Ames has demonstrated DAVE-ML as ready for submittal as the Recommended Practice for the standard
- Submittal to the AIAA to begin momentarily

Variable Names

- Names database and definition complete
- Naming convention taken from STARS Simulation work (Lead by NAWCTSD)
- Short names taken from NASA Ames

Example Table of Names

Symbol	Short Name	Long Name	Same as	Description	Units	Sign	Initial Value	Minimum Value	Maximum Value	Reference	Note	Date Last Changed
	8 Character	33 Character name	STARS?	(including axis system if applicable)		Convention						
	PHI	Euler_Roll_Angle_deg	y	Roll Euler Angle, L (local) Frame	DEG	RWD		-180	180		2	
	THET	Euler_Pitch_Angle_deg	y	Pitch Euler Angle, L (local) Frame	DEG	ANU		-90	90		2	
	PSI	Euler_Yaw_Angle_deg	m	Yaw Euler Angle, L (local) Frame	DEG	ANR		-180	180		2	
	PHIR	Euler_Roll_Angle_rad	y	Roll Angle, L frame	RAD	RWD				10) 1.3.3.3	1,2	
	THETR	Euler_Pitch_Angle_rad	y	Pitch Angle, L frame	RAD	ANU				10) 1.3.3.2	1,2	
	PSIR	Euler_Yaw_Angle_rad	m	Yaw Angle, L frame	RAD	ANR				10) 1.3.3.1	1,2	
	PHID	Euler_Roll_Angle_Rate_ra	y	Euler roll rate, L frame	RAD/SEC	RWD						
	THED	Euler_Pitch_Angle_Rate_	y	Euler pitch rate, L frame	RAD/SEC	ANU						
	PSID	Euler_Yaw_Angle_Rate_r	y	Euler yaw rate, L frame	RAD/SEC	ANR						

AIAA American Institute of Aeronautics and Astronautics

SAIC® From Science to Solutions™ SAIC.COM

Present Status-Data Formats

- Will use Hierarchical Data Format (HDF) as the "Core" format

HDF is a multi-object file format for the transfer of graphical and numerical data between machines. Data models supported include raster images, color palettes, scientific data sets, text entry, binary tables. It was developed by The National Center for Supercomputing Applications (NCSA), located at the University of Illinois at Urbana-Champaign.

- Information from
  - Introduction to HDF5, NCSA/University of Illinois at Urbana-Champaign, May 2000
  - http://hdf.ncsa.uiuc.edu/HDF5/papers/presentations/HDF5_overview
  - Introduction to HDF5 Data Model, Programming Model and Library APIs, NCSA, October 2004
  - http://hdf.ncsa.uiuc.edu/training/hdf5-class/index.html

HDF4 vs HDF5

- HDF4 – Based on Original 1988 version of HDF
  - Backwardly compatible with all earlier versions
  - 6 basic objects
    - Raster image, multidimensional array, palette, group, table, annotation
- HDF5 – First released in 1998
  - New format(s) and library – not compatible with HDF4
  - 2 basic objects

HDF4 Shortcomings

- Limits on object and files size (<2GB)
- Limits on number of objects (<20K)
- Rigid data models
- I/O Performance

New HDF5 Features

- More scalable
  - Larger arrays and files
  - More objects
- Improved data model
  - New data types
  - Single comprehensive dataset object
- Improved software
  - More flexible, robust library
  - More flexible API
  - More I/O options
  - Parallel processing

Example HDF5 File