Validating AnIML Files

Gary W. Kramer

Biochemical Science Division
National Institute of Standards and Technology

NIST

This presentation will:

- Describe XML Conformance Testing
- Discuss Types/Levels of Conformance Testing and Authorities Against Which Conformance is Tested
- Describe the XML Components of AnIML and How They Interact
- Discuss Conformance Testing of AnIML's XML Components
- Describe the NIST AnIML Data File Validator
- Discuss the Requirements for ASTM AnIML File Validator

NIST

Why test XML files for conformance?

- XML is more than a data format, more than a system of identifying content (data elements and attributes) with labels (tags)
- XML provides a means for defining the structure, content and semantics of XML documents. It expresses shared vocabularies and allows machines to carry out rules made by people
- Conformance is testing to determine whether an implemented system fulfills its requirements against a standard or authority.

NIST

XML Conformance Testing
Well-Formedness Checking & Validation

- Well-Formedness Checking is Basically Syntax and Language-Rule Checking
  - All XML documents must be well-formed
  - All mal-formedness errors are fatal
    - When a parser encounters an error, it stops
- Validation is Basically Checking Against the Content Model
  - With validation, all that is not permitted is forbidden
  - Validation errors and warnings are not necessarily fatal
    - The outcome of an error depends on the parser's instructions

NIST

Syntax Guidelines for XML Well-formedness

- Every XML document must have exactly one root element
- Every start tag must have a matching closing tag
- Elements must be properly nested but may not overlap
- Empty elements must be formatted correctly
- Attributes must be enclosed in quotation marks
- An element can not have multiple attributes with the same name
- Comments and processing instructions cannot appear inside tags
- No unescaped < or & characters can occur in character data of an element or attribute
- …

NIST

Naming and Design Rules

- Guidelines and rules for creating schemas to enhance the interoperability of documents created by different markup languages
- Used to enhance the potential interoperability of AnIML files with those of other markup languages such as UBL, UN/CEFACT, GJXDM…
- In AnIML, Technique Definition NDRs are used to enhance the commonality of Technique Definitions across different analytical genre

NIST

XML Schema

- An XML document
- Provides the content model for data being described
- Constrains and governs order and sequence of elements
- Specifies permissible value spaces for all data in content model
- W3C Final Recommendation XML Schema 1.0
- Generally replaced the older, less-flexible Document Type Definition (DTDs) as the XML content model

NIST

XML Instance Document

- An XML document described by a schema
- In the simplest case, an Instance Document is considered valid if it satisfies all the constraints specified by the schema

NIST

AnIML Schema

- Core Schema – a generic data content model useable for describing all/any analytical chemistry data
- Technique Schema – a content model for describing metadata used to constrain the data model provided by the Core Schema to comply with the commonly accepted conventions of a specific analytical technique—an AnIML Technique Definition

NIST

AnIML Technique Definitions

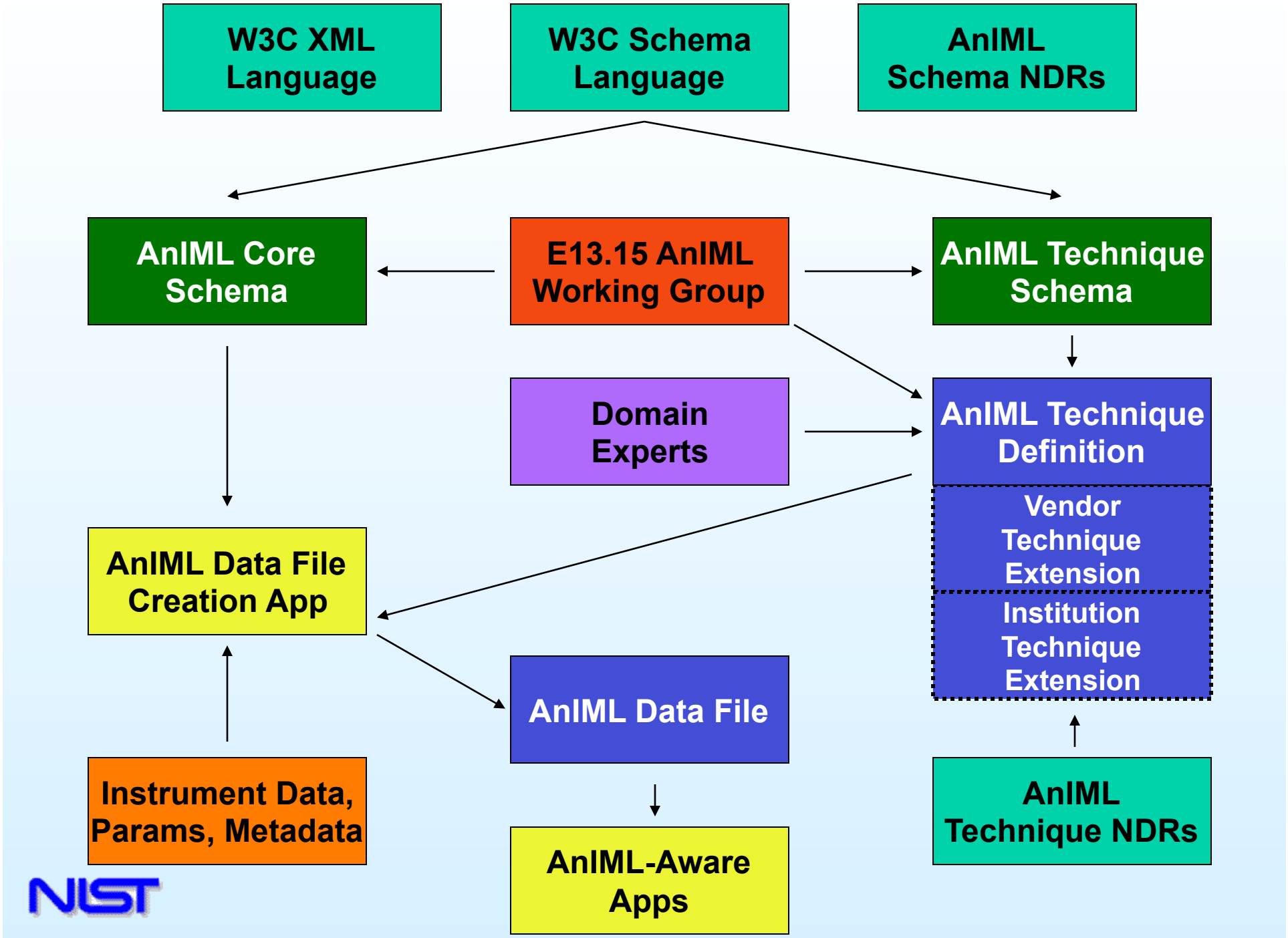
- XML Instance Documents
- Conform to the AnIML Technique Schema
- Purpose is to constrain the data representations in the very flexible AnIML Core to comply with the data formats and metadata conventions commonly accepted by those practicing the technique
- May be appended by AnIML Technique Extensions (provided by vendors, institutions, users…) to add new data representations and metadata (but not to redefine or remove existing elements and attributes of the model)

NIST

AnIML Data Files

- XML Instance Documents
- Content model defined by the AnIML Core Schema as constrained by applicable AnIML Technique Definitions
- Contains the data and metadata from the experiment, measurement, simulation, and/or data processing

NIST


W3C XML Language
W3C Schema Language
AnIML Schema NDRs
AnIML Core Schema
E13.15 AnIML Working Group
AnIML Technique Schema
Domain Experts
AnIML Technique Definition
Vendor Technique Extension
Institution Technique Extension
AnIML Data File Creation App
AnIML Data File
Instrument Data, Params, Metadata
AnIML-Aware Apps
AnIML Technique NDRs
NIST

Types of Validation For Content Models (Schemas)

- Syntactical or Well-Formedness Checking - syntax/ grammar/rule checking against W3C XML language rules and W3C Schema language
- Advanced Compliance Checking - checking against encodable Naming and Design Rules (NDRs)

NIST

Types of Validation
For Instance Documents

- Syntactical or Well-Formedness Checking - syntax/ grammar/rule checking against W3C XML language rules
- Simple Validation - checking for organization, structure, required elements/attributes, data types, … against the content model (schema)
- Advanced Compliance Checking - checking against encodable NDRs (used only for Technique Definitions and Extensions)
- Semantic Validation - checking data content against boundary limits, algorithms, and/or rule bases

NIST

Validation Tools

- Development Tools
  - Parsers
  - Validating Parsers
  - XML Editors
  - XML Viewers
  - XML Validators
  - Web-based vs. Stand-Alone
- Applications
  - Web Browsers (Internet Explorer ≥ vers 5)
  - Office Tools (Microsoft Office 2007 and 2008)
- Note: each validation tool is a little different
  - As yet there is no W3C standard for XML validators
  - W3C XML Schema Group Web-based Validator – XSV
    - Validates Schemas and Instance Documents
    - Work in progress
    - Current XSV version: XSV 3.1-1 as of 12/11/2007

NIST

NIST XML Conformance Tools

- Developed NIST MEL MSID www.mel.nist.gov/msid/XML_testbed/validation.html
- **Business Process Monitor** - provides monitoring and conformance checking capability for choreographed transactions between business partners
- **Content Checker** - provides a collaborative environment for content validation
- **Naming Assister** - improves schema readability and consistency and provides consistent naming conventions for elements and types based on ISO 11179
- **Naming Report** - generates a list of terms used to construct the schema's elements, type, and attributes names
- **NIST XML Schema and Validation Web Services** - a web service that can be used to validate XML schema files remotely
- **Schematron Editor Tool** - java-based GUI tool to create, view, and modify Schematron-constraint-specification files easily for validating XML instances
- **Quality of Design (QOD)** - assists you to use XML schema consistently for the specification of information
- **Quality Measurement Data (QMD**) - provides a development tool to an implementer of the XML-based QMD specification for measurement devices. It will allow end users and tier suppliers to perform their own tests on vendor implementation files easily to verify vendor claims of compliance to QMD

NIST

Validation of AnIML Schemas

- Validate AnIML Core Schema and AnIML Technique Schema
- Can use “standard” XLM validation tools
- Only need to validate AnIML schemas once
- Validation done by ASTM E13.15
- AnIML Requirements Document dictates that AnIML schemas must validate with:
  - Altova XMLSpy 2005 or later
  - Microsoft Visual Studio 2005 or later
    - XML Text Reader - yes
    - XML Document Reader - yes
    - .net dataset parser – not yet

NIST

NIST Validator for AnIML Data Files

- Created by Patrick Gleichmann and Kordian Placzek
- Validates only AnIML Data Files
- Organized into Procedure Blocks (P-blocks) to speed checking by running P-blocks in parallel where possible
- Well-formedness checking using DOM4J
- Validation against AnIML Core Schema using Apache Xerces
- Validation against AnIML Technique Definitions using Java-based P- Blocks
- Limited Semantic Validation using Java-based P-Blocks
- Currently does not validate signatures
- Extensible by creating and linking in new P-Blocks

NIST

Procedure Block Content

- Dependencies – determines if a P-Block can run independently in parallel or must run before/after another P-block
- XPath – used to locate elements to be checked
- Validation method – how to do the checking

NIST

Requirements for the ASTM AnIML File Validator

- Application tool to validate AnIML Technique Definitions, AnIML Technique Definition Extensions, and AnIML Data Files
- Can not use “standard” XML validation tools
- Check XML well-formedness
- Validate AnIML Technique Definitions and AnIML Technique Definition Extensions against AnIML Technique Schema and encodable AnIML Technique Definition NDRs
- Validate AnIML Data Files against AnIML Core Schema and applicable AnIML Technique Definitions with appropriate AnIML Technique Extensions
- Allow checking with and without audit trailing and signatures
- Some simple semantic validation included
  - Bounds checking
  - Limit checking

NIST

Validation Authorities and Levels of Validation for AnIML Documents

- W3C Recommendations
  - XML Language Rules
  - Schema Language Rules
- ASTM AnIML Standards
  - AnIML Core Schema
  - AnIML Technique Definitions
    - Vendor Technique Definition Extensions
    - Institutional Technique Definition Extensions
    - Other Technique Extensions
  - Audit Trailing
  - Digital Signatures (W3C DSIG)
- Institutional Mandates
- Regulatory Agency Requirements

NIST

How Much Conformance Testing is Enough?

- Depends on business case requirements
- Depends on “customer” for data
- Depends on your need for speed verses your need for conformance
- As business needs change, AnIML Data Files may need to be re-validated against higher authorities to assure conformance with requirements
- Must realize the limitations of existing data files
  - Legacy data may not meet modern requirements

NIST

More Information

- ANIML
  - http://animl.sourceforge.net
  - http://www.animl.org
- XML Conformance and Validation
  - Cover Pages
    - http://xml.coverpages.org/xmlConformance.html
    - http://www.oasis-open.org/cover/check-xml.html
  - XML Schema Validation
    - XSV v3.3-1 (12/'07) web service
      - http://www.w3.org/2001/03/webdata/xsv

NIST

Ensuring the Validity of AnIML Components

- Validation can ensure that AnIML documents comply with authorities.
- How do we ensure that AnIML authorities have not been compromised?

NIST

Ensuring that AnIML Schemas are Valid

- Cannot build functional security block into schemas
- Secure Websites, Certificates, and Signature Authorities
- Website-provided file hashes (MD5 and/or SHA-1) in separate, digitally signed files
  - Requires separate application to verify file hash
  - Such applications are available in the web for free

NIST

Ensuring that AnIML Instance Documents are Valid

- Useable for AnIML Technique Definitions, AnIML Technique Extensions, and AnIML Data Files
- Could use same mechanism as that for schemas
- Can build functional security block into instance documents
- Security blocks can be locked with digital signatures
- Security blocks detail authorities used and validation testing details
- Security blocks created by the AnIML validation program
- Provides an authentication credential, so validation need only be done once unless needs validation change
- Validation solution for large AnIML data files

NIST