Versioning and naming XML Schema.

Draft for discussion

Stephen M Richard

January 22, 2012, updated Dec 24, 2012

# Introduction

The US Geoscience Information network is developing specifications for interoperable data services to facilitate the discovery and access to information resource on the World Wide Web. The primary focus of this development has been on developing profiles of existing XML schema (e.g. ISO19139) or in developing schema for XML interchange formats using GML Simple features (e.g. http://geothermaldata.org/page/ngds-content-models). In order for an orderly evolution of the system such that existing applications are not broken, careful consideration and discipline will be necessary in versioning interchange formats developed for USGIN services.

## Change management: Versioning

A USGIN information exchange is based on a content model and an implementation of that content model. Content models may evolve over time, based on changing understanding or user requirements. Multiple implementations of a content model may exist, using different formats e.g. XML, JSON, CSV…, and an implementation of a particular content model in a particular format may be modified (different field order, field names, data types…). In order to document a service to enable software clients to determine that it provides information in a known document format, any interchange document should include an identifier for the interchange format used by that document. The interchange format identifier must map to a particular version of the interchange feature content model, and documents with the same interchange format ID must validate according to a fixed, documented collection of validation rules (xml schema, schematron, other validation process…).

### Content model

The basis for information interchange is a content model that defines feature types and the attributes associated with each feature instance. The content model is a conceptual model for the information associated with a feature. A content model is defined by a specification document; for the NGDS, content models are specified by Microsoft Excel workbooks. The Open Geospatial Consortium calls the content models ‘Abstract Specifications’, and they are specified in text documents. Each content model defines one or more feature types designed for some particular information interchange use cases. A version of a content model is defined by the features it includes, and the collection of fields, cardinality obligations for those fields, and the data types assigned to the fields for each feature type.

A major-minor numbering scheme is used to document content model versions. Major numbers indicate a model based on the same collection of use cases and approach to representation of a feature. Minor versions are incremented when fields are added or removed, obligation rules are changed, or data types are changed.

### Interchange format identifier

In order to be used for information exchange, a content model must be implemented using a computer-processable encoding scheme. USGIN interchange formats are implemented using the eXtended Markup Language (XML); the syntax and structure of interchange documents is specified by an XML schema. A variety of approaches are available to document xml schema versions. USGIN mandates that a namespace URI is defined for each xml schema, and that this URI provides a unique mapping to a particular content model version, and also identifies the implementation version.

Version numbering for implementations follows the major-minor approach used for content models, and in general, the numbering for implementation versions should parallel the numbering for versions of the conceptual model that is implemented. XML, JSON, CSV or other basic implementation formats are distinguished by distinct base URI for implementations using different formats; these are considered separate resources, not versions of a single resource. Minor numbers indicate changes in field order, obligation rules, data type, or mandated syntax for field content. Each version (unique major.minor value) must be associated with a unique validation process.

# Use Cases/scenarios

For some in-depth background on versioning approaches and use cases for XML schema, see W3C discussions:

* Framework for discussion of versioning (XML Schema Working Group, 2005-03-02, accessed at <http://www.w3.org/XML/2004/02/xsdv.html#id2606257>)
* XML Schema Versioning Use Cases (XML Schema Working Group, 2006-01-31, accessed at <http://www.w3.org/XML/2005/xsd-versioning-use-cases/>)

# Recommendations

1. Namespace URI identifies schema and version. URI’s change when non-backward compatible changes are introduced.
2. Major version number is associated with fundamental content model
3. Minor version number increments with non-backward compatible changes
4. Minor-minor version number is encoded in the Schema version and as an attribute of the root element in schema instances, but schema uses same URI.
5. Schema name includes major and minor version number
6. If an element type is changed between versions, the element in the new version should be given a new name.
7. Change the name of an element's type only if its *immediate* content has changed.
8. To conform to rule 6, root element names for interchange packages should include the major and minor version number of the schema that will validate instances.
9. Forward compatibility may be implemented by including an anyURI 0..\* element as the last element in the schema, but for maximum interoperability, best practice is to deploy at least one interchange format output schema that has a fixed set of elements (some of which may be optional).
10. Namespace URI should dereference to normative XML schema for that namespace.
11. Once a service is deployed using a particular schema for the data interchange documents, that service should be maintained online and accessible until it can be determined that there are no users, or that the user community has been notified with sufficient lead time (open to discussion how long that is…) that the service instance will be discontinued.

## Example:

Fragment from normative XML schema:

<xs:schema … targetNamespace="http://www.example.org/schemas/example/1.3" version="1.3.3">

<xs:element name="Example">

<xs:complexType>

…<!-- various elements defined here… -->

<xs:attribute name="schemaVersion" type="xs:decimal" use="required"/>

</xs:complexType>

</xs:element>

<!-- rest of the schema …-->

</xs:schema>

Example root element in an interchange instance document:

<Example schemaVersion="1.3.3" xmlns=" http://www.example.org/schemas/example/1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation=" http://www.example.org/schemas/example/1.3 http://www.example.org/schemas/example1.3.3.xsd">

Note that the schema location in the instance document is a convenience and may point to a cached schema location for various reasons. The namespace URI should always dereference to the current minor-minor version of the schema.