

XML Schema

Charles E Campbell Ph.D.
River City Research Group LLC,
Orangevale, CA
campbelc@calweb.com

Andrew Eisenberg
IBM, Westford, MA 01886
andrew.eisenberg@us.ibm.com

Jim Melton
Oracle, Sandy, UT 84093
jim.melton@acm.org

Guest Column Introduction

This month's column deals with metadata for XML, primarily the W3C's XML Schema Recommendation. XML Schema is often seen as highly complex, but quite powerful.

We have worked with Chuck Campbell in several standards arenas, including the SQL standard and XML Query. Chuck is an invited expert to the W3C's XML Schema WG, so we invited him to write a column outlining the features and futures of XML Schema.

Jim Melton and Andrew Eisenberg

Introduction

We begin by asking the question: What is an XML Schema? In the broad sense, it represents the metadata for the associated XML document or class of XML documents. The Schema describes what can and cannot be included in an XML document.

XML Schema's development began within the W3C, the World Wide Web Consortium, in mid-1998, and development continues today. I did not join the Working Group until the 8th face-to-face meeting in November 1999, which was held in Reston, Virginia.

XML Schema's goal was to extend the capabilities of the DTD (Data Type Definition), which was introduced in XML 1.0 and inherited from SGML. The XML 1.0 Recommendation simplified the DTD over its SGML ancestor. Accessing an SGML document required a DTD, while XML states that a XML parser must be able to read an XML document without a DTD. Unlike the DTD, XML Schemas are written as XML documents, rather than as Extended BNF. More information on DTDs can be found in the XML Recommendation [9].

Both DTDs and XML Schemas are used to validate XML documents and to provide metadata information about those XML documents. XML Schema extends the functionality provided by DTD, most importantly with the addition of a type system.

Two major interest groups, participants representing the document interests and those representing the data interests, make up the XML Schema Working Group (WG); this is much like the composition of the XML Query Working Group. At times, this has been a source of struggle for the Working Group to contain the XML Schema Recommendation's features so that it does not become too complex, while keeping true to the requirements and goals of the Working Group.

This column is just an overview of the W3C XML Schema; there are a number of good web-based tutorials on the subject that you should consider if you want a more in-depth look at the subject.

Web-Based XML Schema Tutorials, Guides, and Presentations

The W3C's official XML Schema Tutorial is actually part of the XML Schema Recommendation [1].

There are also a number of other sources on the subject; these are taken from the XML Schema Public Page at the W3C web site [10] [11] [12] [13] [14] [15] [16] [17] [18] and [19].

The W3C Process

In June 1998, the W3C looked at what work needed to be defined for the continued development of XML. Schema support was discussed as one of the major technical components needed. In July 1998, the XML Activity was approved by the W3C and in August 1998, the XML Schema WG came into being at the XML Activity meeting in Montreal.

In August 1998, a call for participation in the newly formed XML Schema Working Group went out to the W3C Membership. Of the original fifteen-workgroup members listed on the call for participation, six are still participating in the development of the XML Schema technologies as members of the Working Group. The first face-to-face meeting was scheduled for November 1998 in

Chicago. The main topic was the development of the requirements document.

The W3C has established several domains for grouping the development of its Recommendations. The XML Schema Activity is part of the XML Activity and is part of the Architecture Domain.

Within the W3C Process, specifications progress through several stages of maturity:

- Working Draft (WD)
- Last Call Working Draft
- Candidate Recommendation (CR)
- Proposed Recommendation (PR)
- Recommendation (REC)

The XML Schema documents that I discuss here are the public documents that are available on the W3C's public XML Schema Web Site, which can be found at: <http://www.w3.org/XML/Schema>

The W3C prohibits disclosure of Working Group activities until they are made public. However, the public can reach the Working Group members through the XML Public Comments mailing list. The address of that mailing list is: www-xml-schema-comments@w3.org. Each comment received is taken very seriously by the Working Group and becomes a major part of the Working Group's duties. Each comment is addressed by the WG and a response is sent to the comment's author. Many comments have led to changes and addition to the XML Schema Recommendations in the past and will continue to help guide the Working Group in the future.

If you have a vested interest in the development of the XML Schema Recommendations and you are not a member of the W3C, reviewing the public drafts and submitting comments is an excellent way to participate in the XML Schema development process. In addition, there is another public mailing list for XML Schema developers. The XML Schema developer's mailing list is xmlschema-dev@w3.org. You can also subscribe to the W3C's weekly newsletter by sending a message to: <http://lists.w3.org/Archives/Public/w3c-announce/>

There are two other ways of participating in the development of the XML Schema Recommendations. Both require that you be a member of the W3C or that you work for a W3C member company. The Working Group permits up to two members from any W3C member company, a Primary and an Alternate. If you work for a W3C member company, even if your company is not a member of the XML Schema Working Group, you can participate in the XML Schema Interest Group (IG).

The XML Schema Working Group uses the XML Schema IG mailing list to discuss the development work that takes place within the Working Group (WG). This is a very rich mailing list for technical information on the development of

XML Schema Recommendations. The WG mailing list is mostly used for administration matters and does not have much technical content.

The XML Schema WG has been gathering requirements for future versions of XML Schema, but the current version of the Recommendations at the time of the writing of this article is 1.0.

The Set of XML Schema Documents

The XML Schema WG has produced the following documents:

- XML Schema Part 0: Primer [1]
- XML Schema Part 1: Structures [2]
- XML Schema Part 2: Datatypes [3]
- XML Schema Requirements [4]

The normative part of the XML Schema Recommendation is made up of *XML Schema Part 1: Structures*, and *XML Schema Part 2: Datatypes*. *XML Schema Part 0: Primer* is non-normative and provides examples of the use of XML Schema. It is highly recommended that you keep the Primer close to hand when studying the Structures document, which is considered the most difficult to understand.

Several notable documents provided input into the development of XML Schema:

- Jan 2000: *Datatypes for DTDs (DT4DTD) 1.0*, [5]
- Jan '99: *Document Definition Markup Language (DDML) Specification, Version 1.0* submitted to W3C [6]
- Sep '98: *Schema for Object-oriented XML* submitted to W3C [7]
- Jan 1998: XML Data submitted to W3C [8]

Several of these proposals were developed by members of the Working Group and had direct impact on the directions that XML Schema took during its initial development.

Historical Review of XML Schema

There have been several attempts at developing a schema language for XML in the past. Many of these efforts have had an impact on the W3C's Schema efforts. Most of these schema languages are no longer in use, but their contributions are undeniable.

- **XML-Data:** Describes an XML vocabulary for schemas, that is, for defining and documenting object classes. It is a W3C Note dated 5 January 1998 and can be found at: <http://www.w3.org/TR/1998/NOTE-XML-data/>
- **XDR:** External Data Representation Standard, which is now RFC 1014. It was authored by Sun

Microsystems, Inc and others and is dated Jun 1987. XDR is a standard for the description and encoding of data. The RFC can be found at: <http://www.faqs.org/rfcs/rfc1014.html>

- **DCD:** Document Content Description: “This document proposes a structural schema facility, Document Content Description (DCD), for specifying rules covering the structure and content of XML documents. The DCD proposal incorporates a subset of the XML-Data Submission XML-Data and expresses it in a way, which is consistent with the ongoing W3C RDF (Resource Description Framework) effort; in particular, DCD is an RDF vocabulary. DCD is intended to define document constraints in XML syntax; these constraints may be used in the same fashion as traditional XML DTDs. DCD also provides additional properties, such as basic datatypes.” [taken from the abstract]. This is a W3C Note dated 31 July 1998 and can be found at: <http://www.w3.org/TR/NOTE-dcd>
- **SOX:** Schema for Object-Oriented XML 2.0: “SOX is a schema language (or metagrammar) for defining the syntactic structure and partial semantics of XML document types. As such, SOX is an alternative to XML DTDs and can be used to define the same class of document types (with the exception of external parsed entities). However, SOX extends the language of DTDs by supporting: An extensive (and extensible) set of datatypes
 - * Inheritance among element types
 - * Namespaces
 - * Polymorphic content
 - * Embedded documentation
 - * Features to enable robust distributed schema management.All of these features are supported with strong type checking and validation. A SOX schema is also a valid XML instance according to the SOX DTD, enabling the application of XML content management tools to schema management.” [taken from the abstract]. SOX is a W3C Note dated 30 July 1999 and can be located at: <http://www.w3.org/TR/NOTE-SOX/>
- **DDML:** Document Definition Markup Language Specification Version 10. “This document proposes Document Definition Markup Language (DDML), a schema language for XML documents. DDML encodes the logical (as opposed to physical) content of DTDs in an XML document. This allows schema information to be explored and used with widely available XML tools. DDML is deliberately simple, providing an initial base for implementations. While introducing as few complicating factors as

possible, DDML has been designed with future extensions, such as data typing and schema reuse, in mind.” [taken from the abstract]. This document is a W3C Note dated 9 January 1999 and can be located at:

<http://www.w3.org/TR/NOTE-ddml>

Currently, there are several attempts to create an alternative to the W3C’s XML Schema. These are mentioned later.

XML schemas vs. DTDs

First off, lets debunk the myth that XML Schema is meant to replace the DTD. Many users of markup languages have invested heavily in the development of DTDs to help manage their businesses and industries. Many industries have standardized DTDs that manage the XML data between applications. If these work well, then there is no reason to convert to XML Schemas. An XML Schema is more verbose than a DTD that does the same thing. XML parsers support DTDs fully and, for many cases, the DTD may be all that is needed. However, there is very little data typing available when using DTDs, and in cases where data typing is an important issue, then XML Schema provides the better solution.

The other major difference between the DTD and an XML Schema is the fact that the DTD is not an XML document; it is written in Extended BNF notation. By contrast, the XML Schema is itself an XML document. To retain compatibility between XML Schema and XML 1.0 DTDs, the simple types ID, IDREF, IDREFS, ENTITY, ENTITIES, NOTATION, NMTOKEN, and NMTOKENS should be used only in attributes. There are tools (such as XML Spy: <http://xmlschema.xmlspy.com/>) that convert DTDs to XML Schemas, so an investment in DTDs can be leveraged and need not be lost in the conversion process.

In-line XML Schemas

When XML 1.0 arrived, most simple examples used an in-line DTD embedded in the XML document. In other words, the XML document also included the DTD as part of the document. XML 1.0 also supports external DTDs, such that the XML document in one file references the DTD, which is in another file. However, you can look high and low, with little success, to find an example of support for in-line schemas. There is little support outside of Oracle’s XML tools and Microsoft’s .net tools that support the concept of an in-line XML Schema.

The W3C’s XML Schema Working Group is currently composing a Note to address the need to document ways in which an in-line Schema can be

used. The concept of an in-line Schema has never been prohibited in the XML Schema Recommendation, but the topic was also never truly addressed. There are three approaches that can be taken:

- First, the Schema could be required to be the first part of the XML Document.
- Use a specific <schema> element referenced by an anchor.
- Use an IDREF to reference an in-line Schema; however, this would require the use of a DTD as well.

It will be interesting to see which way the industry decides to support the concept of in-line Schemas. If you have an idea which (or all) should be supported, then it would be helpful to hear your opinions on this topic on one of the public mailing lists.

XML Schema Part 0: Primer

This part of the XML Schema Recommendation [1] is non-normative, which means that it is not a part of the recommendation to which an implementation can claim conformance (or, indeed, implement). Again, the normative parts are Parts 1 and 2. David Fallside of IBM authored the Primer. When David joined the Working Group, he decided that he needed to get a handle on the XML Schema Working Drafts. So he took on the development of the Primer as a way to learn the XML Schema technology. What he gave the rest of us is a very useful tool for someone new to the use of XML Schema and a great way to get a good grounding of the normative parts of XML Schema.

The Primer example centers around the PO, Purchase Order, a concept generally well understood by most readers. It is well organized and should be at your side when you tackle the other parts of the Recommendations. It could be improved by expanding its coverage of Part 2, Datatypes. I look forward to the updates that will be included in future versions of this document.

XML Schema Part 1: Structures

There has always been a love-hate relationship with XML Schema for many users. Many have complained that the Recommendation is too complex and difficult to understand. If there is a culprit for this feeling, it is probably due to the Structures document [2]. It should be noted that the XML Recommendation is a 46-page document compared to the Structure document's 159 pages.

During the development of the XML Schema Recommendations, there was a constant tug and pull on the Working Group to continue to add features to

the recommendation (often known as “feature creep”). Everyone had a favorite feature that they could not live without. However, it was decided that the major goal had to be that XML Schemas would have to have the functionality of the DTD, while adding functionality beyond the DTD. So where do you stop? In the end, I am not sure we made any one really happy. The XML Schema Recommendations are a compromise of several interest groups.

The Structures document was reorganized several times to make the organization of the material more approachable. However, it is still considered by most readers to be a difficult document to get their heads around. The reader should have completed the Primer before attempting the Structures document. It is also a good idea to have the Primer close at hand while reviewing the Structures document.

The *XML Schema Part 1: Structures* document is dependent on these other Specifications:

- XML Infoset
- XML Namespaces
- XPath
- XML Schema Part 2: Datatypes

The Structures document defines the XML Schema usage of attributes, elements, and complex type definitions, as well as the constraint and validation rules associated with them. In addition, the use of namespaces allows for the use of mixed vocabularies with elements, attributes, and complex types having the same name. In other words, an XML document can have elements for a vocabulary dealing with travel and use an element <Port> at the same time as a vocabulary dealing with computer hardware that also uses an element with the name <Port>—without having any naming collisions. This in itself is a tremendous advantage over the DTD.

The use of the XML Infoset will prove to be another topic for XML Schemas. In a W3C workshop, the topic of process control was investigated and the XML Infoset looks like an option. There are many other topics in the document, which assures DTD functionality. The datatype DTD compatibility is covered by the next part of the XML Schema Recommendation: Part 2: Datatypes.

XML Schema Part 2: Datatypes

The final part of the XML Schema Recommendation is *XML Schema Part 2: Datatypes* [3] and, like Part 1: Structures, it is normative. This part of the XML Schema Recommendation develops a robust type system that includes:

- Datatype
- Value Space
- Lexical Space
- Facets

- Datatype Dichotomies
- Built-in datatypes
 - Primitive Datatypes
 - Derived Datatypes

The top of the built-in datatypes hierarchy is **anyType**. I will attempt to depict the built-in datatype hierarchy below:

- **All complex types** are derived by extension or restriction.
- **anySimpleType – an ur type**; other simple types are derived by restriction from anySimpleType:

duration	date
time	dateTime
gYearMonth	gYear
gMonthDay	gMonth
gDay	boolean
base64Binary	hexBinary
float	double
anyURI	QName
NOTATION	
decimal [see additional hierarchy below]	string [see additional hierarchy below]

Both **string** and **decimal** have additional datatypes below them, as illustrated below:

- **string**
 - **normalizedString**
 - **token**
 - **language**
 - **Name**
 - **NCName**
 - **ID**
 - **IDREF**
 - **IDREFS**
 - **ENTITY**
 - **ENTITIES**
 - **NMTOKEN**
 - **NMTOKENS**
- **decimal**
 - **integer**
 - **nonPositiveInteger**
 - **negativeInteger**
 - **long**
 - **int**
 - **short**
 - **byte**
 - **nonNegativeInteger**
 - **unsignedLong**
 - **unsignedInt**
 - **unsignedShort**
 - **unsignedByte**
 - **positiveInteger**

All of these datatypes are derived by restriction with the exception of **IDREFS** and **ENTITIES**, which are both derived by list.

There will be more clarification in future versions of the Recommendations when they are published. I should mention that the Schema WG has recently approved XML Schema 1.0 Second Edition documents, which have all the errata applied and can save the reader the jumping from the Recommendation to the errata document.

NIST (National Institute of Standards and Technology) is collecting a set of XML Schema (built-in) Datatype Conformance Tests. <http://www.nist.gov/> gets you to the NIST Web Site, after which you will need to do a search for XML Schema Conformance Tests. You will get a list of literally thousands of tests.

Future Work

The W3C XML process is working towards the publication of a relatively minor revision of the XML Schema Recommendations in the near future. It is difficult for me to give any details on what is coming up in the future, since I am bound by the W3C confidentiality rules. I can say, however, that requirements are already being gathered for future versions of XML Schema; whether there will be a version 1.1 Recommendation or not is not evident, but it seems quite likely that there will be a version 2.0. The XML Schema Working Group is chartered through September 2003. I see this work continuing for the foreseeable future. However, if you'd like to see more of what is to come, you or your company can either join the W3C or become active in the public mailing list I mentioned. This is an exciting technology that has possibilities beyond the document and validation. It can also pass rules between applications such as a dynamic interface contract, but that is a topic for another article.

References

- [1] *XML Schema Part 0: Primer*, David C. Fallside, 2 May, 2001, <http://www.w3.org/TR/xmlschema-0/>
- [2] *XML Schema Part 1: Structures*, Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, 2 May, 2001, <http://www.w3.org/TR/xmlschema-1/>
- [3] *XML Schema Part 2: Datatypes*, Paul V. Biron, Ashok Malhotra, 2 May 2001, <http://www.w3.org/TR/xmlschema-2/>
- [4] *XML Schema Requirements*, W3C Note, Ashok Malhotra, Murray Maloney, 15 February 1999, <http://www.w3.org/TR/NOTE-xml-schema-req>

- [5] *Datatypes for DTDs (DT4DTD) 1.0*, Lee Buck, Charles F. Goldfarb, Paul Prescod, W3C Note, 13 January 2000, <http://www.w3.org/TR/dt4dtd>
- [6] *Document Definition Markup Language (DDML) Specification*, Ronald Bourret, John Cowan, Ingo Macherius, Simon St. Laurent, *Version 1.0* submitted to W3C as a Note, <http://www.w3.org/TR/NOTE-ddml>
- [7] *Schema for Object-oriented XML*, submitted to W3C as a Note, Andrew Davidson, Matthew Fuchs, Mette Hedin, Mudita Jain, Jari Koistinen, Chris Lloyd, Murray Maloney, Kelly Schwarzhof, <http://www.w3.org/TR/NOTE-SOX/>
- [8] *XML Data*, Andrew Layman, Edward Jung, Eve Maler, Henry S. Thompson, Jean Paoli, John Tigue, Norbert H. Mikula, Steve De Rose, submitted to W3C as a Note, <http://www.w3.org/TR/1998/NOTE-XML-data-0105/>
- [9] *Extensible Markup Language (XML) 1.0 (second Edition)*, W3C Recommendation, 6 October 2000: <http://www.w3.org/TR/REC-xml>
- [10] *XML Schema Tutorial*, by Roger L. Costello, September 2001. <http://www.xfront.com/>
- [11] *The XML Schema Specification in Contrast*, by Rick Jelliffe, Academia Sinica Computer Centre. 2000-02-24. <http://www.ascc.net/~ricko/XMLSchemaInContet.html>
This compares XML Schema with XML DTDs, SGML DTDs, HyTime, and Perl regular expressions.
- [12] *The Current State of the Art of Schema Languages for XML*, Rick Jelliffe, presented paper at XML Asia Pacific 2001, Sydney, Australia. A characterization and comment on the XML Schema Language at the end of 2001. (Unfortunately, this presentation is no longer available on-line.)
- [13] Course “*Programming XML in Java*” Web site by John Punin, Autumn 2001. <http://www.cs.rpi.edu/~puninj/XMLJ/>
- [14] *XML Schema a brief introduction*, by Ian Stuart, October 26, 2001. <http://lucas.ucs.ed.ac.uk/xml-schema/>
- [15] *XML Schema Tutorials materials*: Slides, additional materials, by Henry Thompson at XML 99 in Philadelphia, a (GCA Conference) The slides to Henry’s talk can be found at: http://www.oasis-open.org/cover/thompsonSchemaSlides19991220_files/frame.htm and there is additional information at: <http://www.oasis-open.org/cover/thompsonSchemaAdds19991220.html>
- [16] *Using W3C XML Schema*, by Eric van der Vlist, October 17, 2001. <http://www.xml.com/pub/a/2000/11/29/schemas/part1.html>
- [17] *Schemas for XML*, by Norm Walsh, July 1, 1999. <http://www.xml.com/pub/a/1999/07/schemas/index.html>
- [18] Kal Ahmed has created topic maps from the XML Schema family of specifications. The HTML-ized result is now up at <http://www.techquila.com/topicmaps/xmlschema/>
- [19] Danny Vint has created quick reference cards, available at: <http://www.xml.dvint.com/>

Web References

- [1] W3C: <http://www.w3.org>
- [2] XML Schema WG: <http://www.w3.org/XML/Schema>