New XML Validation Technologies in Action

Alex **Brown**

Abstract

This paper is based from a number of real-world XML validation projects, and compares and contrasts the experience 'in the trenches' with the current state of the art in XML validation standards.

Validation is a topic of some controversy in the XML community. While there has been movement from the basic validation offered by XML 1.0 DTD's, there is little consensus on whether that movement has been in the right direction. Two rather different XML schema languages, from W3C (XML Schema Definition Language) and ISO (RELAX NG) are perceived to compete, and continuing ISO and W3C standardisation work is doing nothing to reconcile the differences.

Meanwhile, the emerging practice of XML pipelining holds out the prospect of 'mixing and matching' technologies to arrive at a more complete solution.

This presentation will present some real examples of 'hard case' XML validation problems and suggest that standards development in this space can be analysed and advanced with more clarity if considered within a conceptual framework which defines what validation actually 'is', and which explores validation within the context of 'real world' use cases.

Currently validation tends, in practice, to be a process which incorporates many activities, possibly including parsing, transformation, data binding, pipelining and report generation. The result of a validation process is often poorly specified and becomes effectively dependent on the 'quality of implementation' of validation tools.

A validation model must rule which of these activities are in and out of scope. This presentation will consider the features of existing schema languages and argue that some traditional aspects of validation, carried over from the SGML past, represent a confusion of concerns which lead to a needlessly complex architecture and implementations. Similarly, even some validation features (such as the PSVI) introduced into recent validation languages blur our understanding not just of what validation, but of what XML itself, is.

Users faced with the need to validate XML often have requirements which haven't been addressed by existing standards, and which might even conflict with 'the philosophy of XML' (whatever that may mean). Custom coding or new approaches to validation standards are required to get the job done. The presentation will give examples of such problem cases, and solutions that have been adopted in response.

Table of Contents

1. Introduction

Validation is a topic of some controversy in the XML community. While there has been movement from the basic validation offered by XML 1.0 DTD's, there is little consensus on whether that movement has been in the right direction. Two radically different XML schema languages, from W3C (XML Schema Definition Language) and ISO (RELAX NG) are perceived to compete, and continuing ISO and W3C standardisation work is doing nothing to reconcile the differences.

This paper attempts to stand back from this local tension, and take a more abstract view of what XML validation is, what it needs to be, how and it is applied in practice, and lessons that can be learned from these things.

2. What is Validation?

What do we understand if somebody says of some XML, that it is not ‘valid?’

In XML terms, validation is a slippery concept that ranges between very precise, and generally understood meaning.

The prime precise meaning sense stems from the use of the term 'valid' in the XML 1.0 Recommendation. According to this validation is something carried out by validating processors, those which:

> MUST, at user option, report violations of the constraints expressed by the declarations in the DTD, and failures to fulfill the validity constraints given in this specification.

The W3C Schema specification introduces the concept of 'Schema-valid' XML: another precise use of the term restricted to a specific technical context.

Other senses of the term invalid include other technical ones (if an XML document conforms to a RELAX NG schema, for example it has its own validity), to less precise ones. Non-expert users of XML often call a document that has been rejected by a parser 'invalid' even if the precise cause of fault is not validation-related.

More generally still the informal sense of 'validity' is often most important. A web designer might ask what use his valid page is if it looks wrong; a publisher won't care that their running heads are marked-up as valid XML is they contain a spelling error; a business doesn't care is his account are valid XML is they don't balance.

But to what extend can validity be machine-enforced? And how far is it reasonable to expect any such mechanisms to fall within the remit of XML processing?

3. The Validation Process

Validation is an activity which potentially has many phases.

3.1. Specification

Specification of rules for validation should ideally be conceived in the abstract, and informed by whatever interests the XML user is likely to have (for businesses, this is likely to be commercial interests). Thus it might be said, ‘We need to identify every customer to link with our commerce database’ or ‘we're going to need to develop a website which synchronises documents with their translated equivalents’.

Often the first wrong step for doomed DTD and schemas is that they are conceived in a technical bubble without sufficient exposure to the wider requirements that document models must satisfy.

However, assuming that sufficient is understood about how what is required, the next step is to translate these requirements into a formal specification of the validation rules.

3.2. Application

The second activity is the application of a formal validation specification to an XML document by a validation engine (I shall use this term rather than 'parser' to maintain the distinction between parsing XML and validating it).

In practice this is done in many different ways: for example using command-line tools, or submitting XML via a web form (as with the W3C's HTML validation service).

3.3. Reporting

XML validators emit some kind of report, or generate some kind of programatic event, when they encounter an error in XML they are processing.

The types of report are as varied as the type of application. With command-line tools messages are emitted describing validationfaults found. When using more sophisticated XML editors it is possible to experience validation report more actively, as such editors can constrain the creation of XML dynamically so that it is not possible unknowingly to create invalid XML.

3.4. Interpretation

Once a validator has spoken, it is up to the XML user to fix the problem it reports. Figuring out what is wrong largely depends on the quality of implementation of the validation engine. The precise nature, and precise location of faults are often quite difficult to devine given a validation error message. For example,

> 1036:11:The content of element type "Product" must match "(RecordReference,NotificationType,DeletionCode?,DeletionText?,RecordSourceType?,(RecordSourceIdentifierType,RecordSourceIdentifier)?,RecordSourceName?,(((ISBN,EAN13?,UPC?,PublisherProductNo?,ISMN?,DOI?,ProductIdentifier*)|(EAN13,UPC?,PublisherProductNo?,ISMN?,DOI?,ProductIdentifier*)|(UPC,PublisherProductNo?,ISMN?,DOI?,ProductIdentifier*)|(PublisherProductNo,ISMN?,DOI?,ProductIdentifier*)|(ISMN,DOI?,ProductIdentifier*)|(DOI,ProductIdentifier*)|ProductIdentifier+),Barcode?,ReplacesISBN?,ReplacesEAN13?,ProductForm,ProductFormDetail?,ProductFormFeature*,BookFormDetail*,ProductPackaging?,ProductFormDescription?,NumberOfPieces?,TradeCategory?,ProductContentType?,ContainedItem*,ProductClassification*,(EpubType,EpubTypeVersion?,EpubTypeDescription?,(EpubFormat,EpubFormatVersion?)?,EpubFormatDescription?,(EpubSource,EpubSourceVersion?)?,EpubSourceDescription?,EpubTypeNote?)?,((SeriesISSN?,PublisherSeriesCode?,SeriesIdentifier*,((TitleOfSeries,Title*)|Title+),Contributor*,NumberWithinSeries?,YearOfAnnual?)|Series+|NoSeries)?,((ISBNOfSet?,EAN13OfSet?,ProductIdentifier*,TitleOfSet,SetPartNumber?,SetPartTitle?,ItemNumberWithinSet?,LevelSequenceNumber?,SetItemTitle?)|Set+)?,TextCaseFlag?,(((((DistinctiveTitle,(TitlePrefix,TitleWithoutPrefix)?)|(TitlePrefix,TitleWithoutPrefix)),Subtitle?,TranslationOfTitle?,FormerTitle*,Title*)|Title+),WorkIdentifier*,Website*,(ThesisType,ThesisPresentedTo?,ThesisYear?)?,((Contributor+,ContributorStatement?)|NoContributor)?,(ConferenceDescription|(ConferenceRole?,ConferenceName,ConferenceNumber?,ConferenceDate?,ConferencePlace?)|Conference+)?,((EditionTypeCode*,EditionNumber?,EditionVersionNumber?,EditionStatement?)|NoEdition),ReligiousText?,LanguageOfText*,OriginalLanguage?,Language*,NumberOfPages?,PagesRoman?,PagesArabic?,Extent*,NumberOfIllustrations?,IllustrationsNote?,Illustrations*,MapScale*,(BASICMainSubject,BASICVersion?)?,(BICMainSubject,BICVersion?)?,MainSubject?,Subject*,PersonAsSubject*,CorporateBodyAsSubject*,PlaceAsSubject*,AudienceCode*,Audience*,USSchoolGrade?,InterestAge?,AudienceRange*,AudienceDescription?,Complexity*,Annotation?,MainDescription?,OtherText*,ReviewQuote*,(CoverImageFormatCode,CoverImageLinkTypeCode,CoverImageLink)?,MediaFile*,ProductWebsite*,(PrizesDescription|Prize+)?,ContentItem*,((ImprintName,Imprint*,PublisherName?,Publisher*)|(Imprint+,PublisherName?,Publisher*)|(PublisherName,Publisher*)|Publisher+),CityOfPublication*,CountryOfPublication?,CopublisherName*,SponsorName*,OriginalPublisher?,AnnouncementDate?,TradeAnnouncementDate?,PublicationDate?,(CopyrightStatement|CopyrightYear)?,YearFirstPublished*,(SalesRights,(SalesRights,SalesRights?)?)?,NotForSale*,SalesRestriction*)?,(((((Height,Width?,Thickness?,Weight?)|Weight|Measure+),Dimensions?)|Dimensions))??,(ReplacedByISBN?,ReplacedByEAN13?,AlternativeFormatISBN?,AlternativeFormatEAN13?,AlternativeProductISBN?,AlternativeProductEAN13?,RelatedProduct*,OutOfPrintDate?)?,(SupplyDetail)*?,(PromotionCampaign?,PromotionContact?,InitialPrintRun?,CopiesSold?,BookClubAdoption?)?)".

is the error message produced by the Xerces-J parser when you leave an element out of an instance that should be valid to a well-known ecommerce DTD. Not surprisingly, XML newcomers find this baffling.

The same validity fault generates this terser warning when validated using another parser,

```
Warning: Content model for Product does not allow element ProductForm here in unnamed entity at line 55 char
13 of ...
```

Which gives a more useful indication of where in the XML the fault might lie.

4. Validation or Not?

One special features of DTD and W3C Schema are worthy of special attention when considering what validation 'is'. This can be categorised broadly as 'augmentation'. DTDs have the ability, through attribute defaulting and the #FIXED mechanism, to perform lightweight transformation of content as part of the validation process. W3C Schema also has this ability, but adds the feature of 'infoset augmentation' whereby the XML content becomes decorated with additional information (such as type information).

In the past - before XSLT and other schema languages existed, and while the XML processing model was in its infancy - DTDs were considered by many, including us, to be a suitable means of providing infoset augmentation. But while some - notably the W3C - still think infoset augmentation in schema languages is a good thing, which is why W3C Schemas has the PSVI ("Post Schema Validation Infoset") - a version of the parsed document with extra information added by the schema. Others agree with James Clark's assessment that this is "a catastrophic architectural mistake" (see http://www.cafeconleche.org/quotes2002.html#quote2002December20).

5. Validation in Action

As a company we have been providing validation mechanisms for customers as long as we have been providing XML services. The examples below illustrate how the approach has evolved, and where it may be heading ...

In the beginning we developed, for a number of publishing customers, bespoke tools based around libraries which used the SAX API. Coding in such an environment is slightly mind-bending (as with many XML applications) as the stream-based model of SAX is often at odds with the way that constraints are expressed by customers.

Such an approach does, however, have several big advantages: the first is economy and speed. Because the validation is stream based there is no reequirement to build a large in-memory tree of the XML being processed, meaning that the memory-hogging, slow-running phenomenon that is the DOM can be avoided. Another advantage is that by hosting the validator with a fully-fledged programming language (Java say), there is great flexibility on tap: if a computer can compute it, such validators can apply it.

On the down side, such validators required fairly specialised development skills to develop and maintain. In our experience customers were reluctant to adopt a tool which needed a 'programmer' to look after it.

5.1. XPath

XPath is the basis of XML validation in a number of commercial and free tools. The most well-known language that uses XPath as the basis of its validation rules is Schematron, of which there are a number of implementations. The Schematron language is currently in the latter stages of ISO standardisation as part of DSDL.

The idea behind XPath is simple: it is a technology for selecting nodes within an XML document. Validation rules may be specified by writing expressions which select nodes which are of interest (usually which must, or must *not* exist). A message is then associated with the rule, and a processor will emit the message when the rule is triggered in the course of validation processing.

While XPath offers a very flexible way of specifying validation rules[1], there is a penalty to pay for using XPath: it required in-memory tree to process - for large documents this is slow to build in practice, and obviously the size of document to be validated is constrained by the memory available to use.

There is no doubt the availability of a streaming implementation of XPath would offer a significant boost to its suitability as a basis for validation rule specification.

6. Conclusion

Data validation is a tier of almost every application, and the increasing prevalance of XML as a universal digital information format signals the need for well-defined, preferably standardised, ways of performing the kinds of validation XML users require, in ways which are likelty to benefot those users.

The history of XML validation is one of inadequate technologies, and while emerging technologies and standards hold out the hope of better XML validation mechanisms, the lack of concensus on what validation, as its most basic level, actually 'is' suggests that this is an area in which there is still much to be done.

Bibliography

Clark, James, *RELAX NG and W3C XML Schema*. (Posting to ietf-xml-use; available http://www.imc.org/ietf-xml-use/mail-archive/msg00217.html).

McGrath, Sean, *Modelling with XML - Puzzles or Problems?* (Available http://www.xmluk.org/slides/cambridge-2004/optimal-xml-mcgrath.pdf).

W3C, *XML in 10 points*. (Available http://www.w3.org/XML/1999/XML-in-10-points).

[1] Through extension functions, almost completely flexible

Biography

Alex **Brown**
Technical Director
Griffin Brown Digital Publishing Ltd [http://www.griffinbrown.co.uk/]
Cambridge
United Kingdom

Alex first became interested in structured markup when analysing literary texts for his doctorate (on early Shakespeare editions) in the late 1980s. Following this he worked as a developer on heavily object-oriented C++ application framework for cross-platform multimedia publishing, at the height of the CD-ROM boom. In 1997 Alex was one of the founding directors of Griffin Brown Digital Publishing Ltd, a UK-based company providing XML-based services and products. He is responsible for leading the company's XML consulting and implementation, and his work includes advising clients on XML/IT strategy and practice, mentoring clients' staff, writing DTDs and Schemas, and designing and developing XML software systems in C++, Java and other languages. In 2002, Alex was invited to join the British Standards Institute (BSI) Technical Committee IST/41, where he contributes to ISO/IEC JTC1/SC34 in its formation of the DSDL ISO standard, among other things. Alex writes and speaks regularly on structured markup technologies and their application to information management.