Infinite Diversity in Infinite Combinations

why one schema language is not enough

John Cowan

Copyright

- Copyright 2004 John Cowan
- Licensed under the GNU General Public License
- Black and white for readability
- ABSOLUTELY NO WARRANTIES; USE AT YOUR OWN RISK
- The Gentium font is available at *http://www.sil.org/~gaultney/gentium*

The status quo

- W3C's XML Schema (WXS) is everywhere:
  - Many document types are defined by it
  - SOAP encoding uses its datatypes
  - WSDL descriptions specify message formats with it
  - XPath 2.0 and XQuery 1.0 have it embedded into them

But...

- WXS is difficult
- WXS is incomplete
- WXS is not formally specified
- WXS is not extensible
- There are many useful concepts that WXS can't specify completely or at all

RELAX NG

- An evolution/generalization of DTDs
- Shares the same basic paradigm
- Based on experience with SGML, XML
- Adds and subtracts features from DTDs
- DTDs can be automatically converted

Reusable Knowledge

- Experts in designing other kinds of schemas will find their skills transfer easily to RELAX NG
- Design patterns commonly used in XML DTDs, as well as in WXS, can be reused in RELAX NG
- RELAX NG is much more mature than if based on a completely new and different paradigm
- A much higher degree of confidence in its design is possible

Simplicity

- RELAX NG is designed for simplicity
  - Read it in a few minutes, write it in a few hours
  - Arbitrary restrictions are very few
  - Constructs mean what you would guess they mean
- WXS is full of surprises and corner cases
  - Child elements and attributes not treated uniformly
  - Simple declarations and wildcards are different

Clarity

- The descriptions of RELAX NG are clear
  - The formal definition is complete and normative, with a solid basis in mathematical theory
  - The informal (but normative) prose and the non-normative tutorials are simple and cover the whole language

Clarity

- The WXS normative definition is extremely hard to follow
  - You have to understand it to be *sure* you fully understand a particular WXS schema
  - Most people only understand certain stereotyped ways of using WXS
  - Independent implementations aren't always consistent in their interpretation of the definition

Co-occurrence constraints

- It's common to want to constrain the attributes and child elements of an element jointly:
  - two attributes (or an attribute and a child element) may be mutually exclusive
  - the content model may depend on the particular value of an attribute (like HTML's input element)

Co-occurrence constraints

- WXS doesn't do co-occurrence constraints, even though they come up all the time in real-world documents
- One exception: WXS can specify that an element is nil by using a reserved attribute, `xsi:nil`.
- RELAX NG can handle this and other cases without any special machinery.

Unordered content

- DTDs only support totally unstructured unordered content
- WXS adds the ability to say “all of these children in any order” but only at the top level of an element
- RELAX NG allows unordered content at any level
- RELAX NG allows interleaved content streams

Unordered content

- So this schema ...

```
element head {
  element meta { empty }* &
  element title { text }
  }
```

- ... matches a head element that has any number of meta child elements (including zero) and a required title child element *mixed in anywhere.*

Datatypes

- DTDs have only trivial datatypes for attribute values, none for character content
- WXS has a huge list of badly designed datatypes
- RELAX NG can support the WXS datatypes or any other set (generic or application specific) that can be devised
- An interface standard for pluggable datatype libraries has been defined

WXS Datatype Issues

- Too many primitive types, some duplicative
- Not enough useful primitive types
- No support for localized types
- No structured types except dates and URIs
- No way to validate against an external list
- No way to change anything for yourself

Vocabularies vs. Documents

- WXS specifies a vocabulary of elements and attributes that may appear in documents
- RELAX NG specifies the structure of particular document types
- Namespace Routing Language (not yet standardized) lets you validate composite documents using multiple schemas (including WXS, RNG, and even Schematron)

Finding Schemas

- WXS provides a schemaLocation attribute so that a document can point to its schema, but it's just a hint
- RELAX NG allows any document to be validated against any schema
  - There is no One True Way to find the schema
  - A document can be formatted against different schemas for different purposes

Defaults

- DTDs allow attribute defaulting
- WXS extends this to element content defaulting
- Both mechanisms change the basic content of a document based on whether it's been validated or not
- RELAX NG leaves such transformations up to transformation programs like XSLT, improving modularity

Standards

- WXS is a W3C Recommendation
- RELAX NG is an OASIS Committee Specification and a Draft International Standard
- De facto, WXS has more tools supporting it
- RELAX NG is beginning to spread to those tools, and has some tools of its own
- Even some W3C documents use RELAX NG to specify document types

RELAX NG advantages

- Organized around patterns
- Patterns are composable without restrictions (very nearly)
- Unrestricted support for mixed content
- Unrestricted support for unordered content
- A compact syntax for authoring, an XML syntax for processing

RELAX NG doesn't do ...

- Identity constraints other than ID/IDREF/IDREFS
  - but in WXS they are specified by a specialized sublanguage
- Infoset augmentation
- Inheritance of complex types
- PSVI
  - most WXS validators don't expose it anyway

RELAX NG Tools

- Validators in Java, VB6, C#
- Translators to and from other schema languages, including WXS
  - You can author in RELAX NG and deliver in WXS
- Data binding tools generate object trees corresponding to particular schemas
- Guided editing with XML tools or Emacs

And now, to relax!

MORE INFORMATION

http://www.relaxng.org

http://www.ccil.org/~cowan/idic{.ppt,.odp,.pdf}