XML Domain Tests

Contents

XML Domain Test (DTD)

numcharref	`
.`
parameterentity	p.content
rngpattern	`p.content`
textentity	hi-d-att
xmlatt	`@class`
xmlelement	foreign
xmlnsname	`http://www.w3.org/2000/svg`
xsdattgroup	`p.attributes`
xsdcomplextype	`p.class`
xsdgroup	`p.content`
xsdsimpletype	`type-tmtype-att.class`

XML Domain Test (RNC)

numcharref	`
.`
parameterentity	p.content
rngpattern	`p.content`
textentity	hi-d-att
xmlatt	`@class`
xmlelement	foreign
xmlnsname	`http://www.w3.org/2000/svg`
xsdattgroup	`p.attributes`
xsdcomplextype	`p.class`
xsdgroup	`p.content`
xsdsimpletype	`type-tmtype-att.class`

XML Domain Test (XSD)

numcharref	`
.`
parameterentity	p.content
rngpattern	`p.content`
textentity	hi-d-att
xmlatt	`@class`
xmlelement	foreign
xmlnsname	`http://www.w3.org/2000/svg`
xsdattgroup	`p.attributes`
xsdcomplextype	`p.class`
xsdgroup	`p.content`
xsdsimpletype	`type-tmtype-att.class`

DITA 1.3 proposed feature #13035

Provide "XML mention" domain for marking up mentions of XML syntactic components

Date and version information

- Proposal Completed: 1 Oct 2012
- Change history:
  - 14 Oct 2012:
    - Resolved TBDs.
    - Changed name of "parment" to "parameterentity" per discussion on TC mailing list.
    - Changed name of "textent" to "textentity" per discussion on TC mailing list.
    - Added sample CSS style sheet.
    - Added trailing ";" to text entity, parameter entity, and numeric character reference mentions.
  - 16 Oct 2012:
    - Changed "xmlelem" to "xmlelement" per TC discussion.
- Champion: Eliot Kimber,
- Email discussion: None.

Original requirement

Any documentation that talks about XML vocabularies, including the DITA standard itself, can benefit from dedicated markup for identifying mentions of of XML components.

Use cases

Any documentation that discusses XML vocabulary, including document type reference information, the DITA standard, any other XML application documentation, XML technology instructional material, etc., documentation of XML-based configuration files for software and hardware, etc.

Benefits

- Who benefits: DITA users who create documents about XML markup.
- Expected benefit: Enables search and retrieval of XML markup, automatic styling of mentions to reflect consistent style rules, automatic indexing of mentioned components (element types, attributes, parameter entities, etc.).
- Potential users: Difficult to quantify.
- Degree of positive impact: Moderate-to-significant. Makes documenting XML vocabularies much more convenient and consistent.

Costs

Costs:

- Maintainers of the DTDs and XSDs: Adds a new vocabulary module, which must be integrated into the appropriate shell document types.
- Editors of the DITA specification:
  - How many new topics will be required? 11 new reference topics.
  - How many existing topics will need to be edited?
    - Topic "Domain elements" in the Language Reference will need to reflect this domain.
  - Will the feature require substantial changes to the information architecture of the DITA specification? No architectural change.

- Vendors of tools: Processors that render DITA content visually should provide appropriate styling of each type of mention, e.g., angle brackets around tagnames, "@" before attribute names, etc. In general the implementation requirements are minimal.
- DITA community-at-large. Will this feature add to the perception that DITA is becoming too complex? Will it be simple for end users to understand?

  This feature adds a new optional vocabulary module. Users who need it will appreciate having it readily available. Users who do not need it may safely ignore it. The general architecture and semantics of DITA are not affected by this proposal.

Technical requirements

Define a new vocabulary module, xmlDomain, that defines the following element types:

numcharref

Mention of a numeric character reference, e.g.:

```
<numcharref>x0a</numcharref> is a
 newline character.
```

A typical rendition would be add the "&#" leading characters and ";" trailing character: "
"

parameterentity

Mention of a parameter entity reference, e.g.:

```
<parameterentity>p.content<parameterentity>
 defines the content model
for <xmlelement>p</xmlelement>.
```

A typical rendition would be to add the leading "%" character and ";" trailing character: "%p.content;".

rngpattern

Mention of a RelaxNG named pattern, e.g.:

```
The pattern <rngpattern>anySvg</
rngpattern> allows any SVG element.
```

textentity

A mention of a text entity, e.g.:

```
The <textentity>hi-d-att</
textentity> entity holds the
 contribution for
the <xmlatt>domains</xmlatt>
 attribute.
```

A typical rendition would be to add the leading "&" character and ";" trailing character: "&hi-d-att;".

xmlatt

Mention of an XML attribute, e.g.:

```
The <xmlatt>id</xmlatt> attribute
 may be specified on almost any
element in DITA.
```

A typical rendition would be to add a leading "@" sign: "@id".

xmlelement

Mention of an XML element type, e.g.:

```
The <xmlelement>foreign</xmlelement>
 is used to hold non-DITA markup.
```

A typical rendition would be to add the left and right angle brackets: "<foreign>".

xmlnsname

A mention of an XML namespace name, e.g.:

```
The SVG vocabulary uses the
 namespace
<xmlnsname>http://www.w3.org/2000/
svg</xmlnsname>
```

xsdattgroup

Mention of a named attribute group within an XML Schema document, e.g.:

```
The attribute
 group <xsdattgroup>p.attributes</
xsdattgroup>
defines the attributes for the
 <xmlelement>p</xmlelement> element.
```

xsdcomplextype

Mention of a named complex type within an XML Schema document, e.g.:

```
The complex
 type <xsdcomplextype>state.class</
xsdcompextype> defines
the base content model for the
 <xmlelement>state</xmlelement>
 element.
```

xsdgroup

Mention of a named group within an XML Schema document, e.g.:

```
The group <xsdgroup>text.content</
xsdgroup> defines the full
content model for the
 <xmlelement>text</xmlelement>
 element.
```

xsdsimpletype

Mention of a simple type (datatype) within an XML Schema document, e.g.:

```
The simple type <xsdsimpletype>type-
tmtype-att.class</xsdsimpletype>
defines the allowed values for the
 <xmlatt>tmtype</xmlatt> attribute.
```

xmlDomain.ent:

```
<!-- ============================================================

     XML construct domain

     Provides phrase-level elements for identifying mentions of
```

```
     XML constructs: element types, attributes, etc., as well as
     named constructs used in the main XML document grammar and
     constraint languages (DTD, XSD, and RelaxNG).

     Copyright (c) 2012 OASIS Open


     ============================================================= -->


<!-- ============================================================= -->
<!--                    XML DOMAIN ENTITIES                        -->
<!-- ============================================================= -->

<!ENTITY % xml-d-keyword
  "numcharref |
   parameterentity |
   rngpattern |
   textentity |
   xmlatt |
   xmlelement |
   xmlnsname |
   xsdattgroup |
   xsdcomplextype |
   xsdgroup |
   xsdsimpletype
  "
  >

<!ENTITY   xml-d-att     "(topic xml-d)"
>


<!-- ================== End XML Domain Entities ============ -->
```

xmlDomain.mod:

```
<!-- ============================================================

     XML construct domain

     Provides phrase-level elements for identifying mentions of
     XML constructs: element types, attributes, etc.

     Copyright (c) 2009, 2010 DITA For Publishers

     This domain module may be used by anyone without restriction.

     ============================================================= -->

<!-- ============================================================= -->
<!--                    ELEMENT NAME ENTITIES                      -->
<!-- ============================================================= -->

<!ENTITY % numcharref
  "numcharref"
>
<!ENTITY % parameterentity
  "parameterentity"
>
<!ENTITY % rngpattern
  "rngpattern"
>
```

```
<!ENTITY % textentity
  "textentity"
>
<!ENTITY % xmlatt
  "xmlatt"
>
<!ENTITY % xmlelement
  "xmlelement"
>
<!ENTITY % xmlnsname
  "xmlnsname"
>
<!ENTITY % xsdattgroup
  "xsdattgroup"
>
<!ENTITY % xsdcomplextype
  "xsdcomplextype"
>
<!ENTITY % xsdgroup
  "xsdgroup"
>
<!ENTITY % xsdsimpletype
  "xsdsimpletype"
>

<!-- ============================================================= -->
<!--                    ELEMENT DECLARATIONS                       -->
<!-- ============================================================= -->

<!--                    LONG NAME: XML Element                          --
>
<!ENTITY % xmlelement.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % xmlelement.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT xmlelement %xmlelement.content; >
<!ATTLIST xmlelement %xmlelement.attributes; >

<!--                    LONG NAME: XML Attribute
 -->
<!ENTITY % xmlatt.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % xmlatt.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
```

```
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT xmlatt %xmlatt.content; >
<!ATTLIST xmlatt %xmlatt.attributes; >

<!--                    LONG NAME: Text entity -->
<!ENTITY % textentity.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % textentity.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT textentity %textentity.content; >
<!ATTLIST textentity %textentity.attributes; >

<!--                    LONG NAME: Parameter entity -->
<!ENTITY % parameterentity.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % parameterentity.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT parameterentity %parameterentity.content; >
<!ATTLIST parameterentity %parameterentity.attributes; >

<!--                    LONG NAME: Numeric character reference -->
<!ENTITY % numcharref.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % numcharref.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
```

```
<!ELEMENT numcharref %numcharref.content; >
<!ATTLIST numcharref %numcharref.attributes; >

<!--                    LONG NAME: RelaxNG Named Pattern -->
<!ENTITY % rngpattern.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % rngpattern.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT rngpattern %rngpattern.content; >
<!ATTLIST rngpattern %rngpattern.attributes; >

<!--                    LONG NAME: XML Namespace name ("Namespace URI") -->
<!ENTITY % xmlnsname.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % xmlnsname.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT xmlnsname %xmlnsname.content; >
<!ATTLIST xmlnsname %xmlnsname.attributes; >

<!--                    LONG NAME: XSD Attribute Group -->
<!ENTITY % xsdattgroup.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % xsdattgroup.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT xsdattgroup %xsdattgroup.content; >
<!ATTLIST xsdattgroup %xsdattgroup.attributes; >
```

```
<!--                    LONG NAME: XSD Complex Type-->
<!ENTITY % xsdcomplextype.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % xsdcomplextype.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT xsdcomplextype %xsdcomplextype.content; >
<!ATTLIST xsdcomplextype %xsdcomplextype.attributes; >

<!--                    LONG NAME: XSD Group -->
<!ENTITY % xsdgroup.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % xsdgroup.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT xsdgroup %xsdgroup.content; >
<!ATTLIST xsdgroup %xsdgroup.attributes; >

<!--                    LONG NAME: XSD Simple Type-->
<!ENTITY % xsdsimpletype.content
"
  (#PCDATA |
   %keyword; |
   %text;)*
">
<!ENTITY % xsdsimpletype.attributes
"
  %univ-atts;
  keyref
    CDATA
    #IMPLIED
  outputclass
    CDATA
    #IMPLIED
">
<!ELEMENT xsdsimpletype %xsdsimpletype.content; >
<!ATTLIST xsdsimpletype %xsdsimpletype.attributes; >


<!-- ============================================================= -->
<!--                    SPECIALIZATION ATTRIBUTE DECLARATIONS      -->
<!-- ============================================================= -->
```

```
  <!ATTLIST numcharref  %global-atts;  class CDATA "+ topic/keyword xml-d/
numcharref "  >
  <!ATTLIST parameterentity %global-atts;  class CDATA "+ topic/keyword xml-
d/parameterentity "  >
  <!ATTLIST rngpattern  %global-atts;  class CDATA "+ topic/keyword xml-d/
rngpattern "  >
  <!ATTLIST textentity  %global-atts;  class CDATA "+ topic/keyword xml-d/
textentity "  >
  <!ATTLIST xmlnsname   %global-atts;  class CDATA "+ topic/keyword xml-d/
xmlnsname "  >
  <!ATTLIST xmlatt      %global-atts;  class CDATA "+ topic/keyword xml-d/
xmlatt "  >
  <!ATTLIST xmlelement     %global-atts;  class CDATA "+ topic/keyword xml-
d/xmlelement "  >
  <!ATTLIST xsdattgroup %global-atts;  class CDATA "+ topic/keyword xml-d/
xsdattgroup "  >
  <!ATTLIST xsdcomplextype %global-atts;  class CDATA "+ topic/keyword xml-
d/xsdcomplextype "  >
  <!ATTLIST xsdgroup    %global-atts;  class CDATA "+ topic/keyword xml-d/
xsdgroup "  >
  <!ATTLIST xsdsimpletype %global-atts;  class CDATA "+ topic/keyword xml-d/
xsdsimpletype "  >


<!-- ================== DITA Highlight Domain ==================== -->
```

Figure 1: DTD Syntax domain module declarations

xmlDomainMod.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <!-- DITA 1.3 XML Domain -->

    <xs:annotation>
      <xs:appinfo>
        <dita:domainsModule xmlns:dita="http://dita.oasis-open.org/
architecture/2005/">(topic xml-d)</dita:domainsModule>
      </xs:appinfo>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>


  <xs:group name="xml-d-keyword">
    <xs:choice>
      <xs:element ref="numcharref"/>
      <xs:element ref="parameterentity"/>
      <xs:element ref="rngpattern"/>
      <xs:element ref="textentity"/>
      <xs:element ref="xmlatt"/>
      <xs:element ref="xmlelement"/>
      <xs:element ref="xmlnsname"/>
      <xs:element ref="xsdattgroup"/>
      <xs:element ref="xsdcomplextype"/>
      <xs:element ref="xsdgroup"/>
      <xs:element ref="xsdsimpletype"/>
```

```
    </xs:choice>
  </xs:group>


  <xs:group name="numcharref.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="parameterentity.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="rngpattern.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="textentity.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="xmlatt.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="xmlelement.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
<xs:group name="xmlnsname.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="xsdattgroup.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="xsdcomplextype.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
```

```
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="xsdgroup.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="xsdsimpletype.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="keyword.content"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>

  <!-- Basic form: Numeric character reference, e.g. &#10; or &#x0a; -->
  <xs:element name="numcharref">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="numcharref.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/
numcharref "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="numcharref.class" mixed="true">
        <xs:sequence>
          <xs:group ref="numcharref.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="numcharref.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="numcharref.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="parameterentity">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="parameterentity.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/
parameterentity "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="parameterentity.class" mixed="true">
        <xs:sequence>
```

```
          <xs:group ref="parameterentity.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="parameterentity.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="parameterentity.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="rngpattern">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="rngpattern.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/
rngpattern "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="rngpattern.class" mixed="true">
        <xs:sequence>
          <xs:group ref="rngpattern.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="rngpattern.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="rngpattern.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="textentity">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="textentity.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/
textentity "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="textentity.class" mixed="true">
        <xs:sequence>
          <xs:group ref="textentity.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="textentity.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="textentity.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
```

```
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="xmlatt">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="xmlatt.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/xmlatt "/
>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="xmlatt.class" mixed="true">
        <xs:sequence>
          <xs:group ref="xmlatt.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="xmlatt.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="xmlatt.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="xmlelement">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="xmlelement.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/xmlelem
 "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="xmlelem.class" mixed="true">
        <xs:sequence>
          <xs:group ref="xmlelem.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="xmlelem.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="xmlelem.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="xmlnsname">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
```

```
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="xmlnsname.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/xmlnsname
 "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="xmlnsname.class" mixed="true">
        <xs:sequence>
          <xs:group ref="xmlnsname.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="xmlnsname.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="xmlnsname.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="xsdattgroup">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="xsdattgroup.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/
xsdattgroup "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="xsdattgroup.class" mixed="true">
        <xs:sequence>
          <xs:group ref="xsdattgroup.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="xsdattgroup.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="xsdattgroup.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="xsdcomplextype">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="xsdcomplextype.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/
xsdcomplextype "/>
        </xs:extension>
      </xs:complexContent>
```

```
    </xs:complexType>
  </xs:element>
  <xs:complexType name="xsdcomplextype.class" mixed="true">
        <xs:sequence>
          <xs:group ref="xsdcomplextype.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="xsdcomplextype.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="xsdcomplextype.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="xsdgroup">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="xsdgroup.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/xsdgroup
 "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="xsdgroup.class" mixed="true">
        <xs:sequence>
          <xs:group ref="xsdgroup.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="xsdgroup.attributes"/>
  </xs:complexType>

  <xs:attributeGroup name="xsdgroup.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>

<xs:element name="xsdsimpletype">
    <xs:annotation>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="true">
      <xs:complexContent>
        <xs:extension base="xsdsimpletype.class">
          <xs:attribute ref="class" default="+ topic/keyword xml-d/
xsdsimpletype "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="xsdsimpletype.class" mixed="true">
        <xs:sequence>
          <xs:group ref="xsdsimpletype.content"/>
        </xs:sequence>
    <xs:attributeGroup ref="xsdsimpletype.attributes"/>
  </xs:complexType>
```

```
  <xs:attributeGroup name="xsdsimpletype.attributes">
    <xs:attributeGroup ref="global-atts"/>
    <xs:attributeGroup ref="univ-atts"/>
    <xs:attribute name="outputclass" type="xs:string"/>
  </xs:attributeGroup>


</xs:schema>
```

Figure 2: XSD domain module declarations

Note: These declarations are not complete. Waiting to work out details of how to declare equivalent of <xs:any> in RNC syntax.

xmlDomainMod.rnc:

```
datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"

# ================================================================
# MODULE:    DITA XML Domain - RNC
# VERSION:   1.3
# DATE:      October 2012
# ================================================================


namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"

# Define the domain values of this module
domains-atts-value |= "(topic xml-d)"
# Define domain extension patterns
xml-d-keyword =
   numcharref.element |
   parameterentity.element |
   rngpattern.element |
   textentity.element |
   xmlatt.element |
   xmlelement.element |
   xmlnsname.element |
   xsdattgroup.element |
   xsdcomplextype.element |
   xsdgroup.element |
   xsdsimpletype.element

# Extend the patterns with the domain contribution
keyword |= xml-d-keyword
# Define elements content and attributes

# LONG NAME: Numeric character reference (&#10;, &#x0a;)
numcharref.content =
  (text | \text | tm)*
numcharref.attributes =
  univ-atts,
  attribute outputclass { text }?
numcharref.element =

  element numcharref {
    numcharref.attlist,
    numcharref.content
   }
numcharref.attlist &= numcharref.attributes
```

```
# LONG NAME: Parameter entity reference (%p.content;)
parameterentity.content =
  (text | \text | tm)*
parameterentity.attributes =
  univ-atts,
  attribute outputclass { text }?
parameterentity.element =

  ## The parameterentity (<parameterentity>) element is used to identify
  ## mentions of XML DTD parameter entities.
  ##         Category: Mention elements
  element parameterentity {
    parameterentity.attlist,
    parameterentity.content
   }
parameterentity.attlist &= parameterentity.attributes


# LONG NAME: Relax NG named pattern
rngpattern.content =
  (text | \text | tm)*
rngpattern.attributes =
  univ-atts,
  attribute outputclass { text }?
rngpattern.element =

  ## The rngpattern (<rngpattern>) element is used to
  ## identify mentions of named patterns.
  ##         Category: Mention elements
  element rngpattern {
    rngpattern.attlist,
    rngpattern.content
   }
rngpattern.attlist &= rngpattern.attributes


# LONG NAME: Text entity (&prodname;)
textentity.content =
  (text | \text | tm)*
textentity.attributes =
  univ-atts,
  attribute outputclass { text }?
textentity.element =

  ## The textentity (<textentity>) element is used to identify
  ## mentions of XML text entities.
  ##         Category: Mention elements
  element textentity {
    textentity.attlist,
    textentity.content
   }
textentity.attlist &= textentity.attributes


# LONG NAME: XML attribute
xmlatt.content =
  (text | \text | tm)*
xmlatt.attributes =
  univ-atts,
  attribute outputclass { text }?
xmlatt.element =

  ## The xmlatt (<xmlatt>) element is used to identify
  ## mentions of XML attributes.
```

```
  ##         Category: Mention elements
  element xmlatt {
    xmlatt.attlist,
    xmlatt.content
   }
xmlatt.attlist &= xmlatt.attributes


# LONG NAME: XML element
xmlelement.content =
  (text | \text | tm)*
xmlelement.attributes =
  univ-atts,
  attribute outputclass { text }?
xmlelement.element =

  ## The xmlelement (<xmlelement>) element is used to identity mentions
  ## of XML element types.
  ##         Category: Mention elements
  element xmlelement {
    xmlelement.attlist,
    xmlelement.content
   }
xmlelement.attlist &= xmlelement.attributes


# LONG NAME: XML namespace name (aka "namespace URI")
xmlnsname.content =
  (text | \text | tm)*
xmlnsname.attributes =
  univ-atts,
  attribute outputclass { text }?
xmlnsname.element =

  ## The xmlnsname (<xmlnsname>) element is used to identify
  ## mentions of XML namespace names, e.g. the value used
  ## in @xmlns attributes.
  ##         Category: Mention elements
  element xmlnsname {
    xmlnsname.attlist,
    xmlnsname.content
   }
xmlnsname.attlist &= xmlnsname.attributes


# LONG NAME: XSD attribute group
xsdattgroup.content =
  (text | \text | tm)*
xsdattgroup.attributes =
  univ-atts,
  attribute outputclass { text }?
xsdattgroup.element =

  ## The xsdattgroup (<xsdattgroup>) element is used to identify
  ## mentions of XSD attribute groups.
  ##         Category: Mention elements
  element xsdattgroup {
    xsdattgroup.attlist,
    xsdattgroup.content
   }
xsdattgroup.attlist &= xsdattgroup.attributes


# LONG NAME: XSD complex type
```

```
xsdcomplextype.content =
  (text | \text | tm)*
xsdcomplextype.attributes =
  univ-atts,
  attribute outputclass { text }?
xsdcomplextype.element =

  ## The xsdcomplextype (<xsdcomplextype>) element is used to identify
  ## mentions  of XSD complex types.
  ##         Category: Mention elements
  element xsdcomplextype {
    xsdcomplextype.attlist,
    xsdcomplextype.content
   }
xsdcomplextype.attlist &= xsdcomplextype.attributes


# LONG NAME: XSD group
xsdgroup.content =
  (text | \text | tm)*
xsdgroup.attributes =
  univ-atts,
  attribute outputclass { text }?
xsdgroup.element =

  ## The xsdgroup (<xsdgroup>) element is used to identify
  ## mentions of XSD groups (element groups, as opposed to
  ## attribute groups).
  ##         Category: Mention elements
  element xsdgroup {
    xsdgroup.attlist,
    xsdgroup.content
   }
xsdgroup.attlist &= xsdgroup.attributes


# LONG NAME: XSD simpletype
xsdsimpletype.content =
  (text | \text | tm)*
xsdsimpletype.attributes =
  univ-atts,
  attribute outputclass { text }?
xsdsimpletype.element =

  ## The xsdsimpletype (<xsdsimpletype>) element is used to identify
  ## mentions of XSD simple types ("data types").
  ##         Category: Mention elements
  element xsdsimpletype {
    xsdsimpletype.attlist,
    xsdsimpletype.content
   }
xsdsimpletype.attlist &= xsdsimpletype.attributes


# Specialization attributes. Global attributes and class defaults
numcharref.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/numcharref " ] attribute class
 { text }?
parameterentity.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/parameterentity " ] attribute class
 { text }?
rngpattern.attlist &=
```

```
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/rngpattern " ] attribute class
 { text }?
textentity.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/textentity " ] attribute class
 { text }?
xmlatt.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/xmlatt " ] attribute class { text }?
xmlelement.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/xmlelement " ] attribute class
 { text }?
xmlnsname.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/xmlnsname " ] attribute class
 { text }?
xsdattgroup.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/xsdattgroup " ] attribute class
 { text }?
xsdcomplextype.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/xsdcomplextype " ] attribute class
 { text }?
xsdgroup.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/xsdgroup " ] attribute class
 { text }?
xsdsimpletype.attlist &=
  global-atts,
  [ a:defaultValue = "+ topic/ph xml-d/xsdsimpletype " ] attribute class
 { text }?
```

Figure 3: RelaxNG Compact domain module declarations

Examples

Sample CSS style sheet for XML domain elements

```
/*
==============================
    XML elements. [xml-d]
==============================
*/

*[class~="xml-d/xmlelem"],
*[class~="xml-d/xmlatt"],
*[class~="xml-d/xmlnsname"],
*[class~="xml-d/parameterentity"],
*[class~="xml-d/textentity"],
*[class~="xml-d/numcharref"]
*[class~="xml-d/rngpattern"]
*[class~="xml-d/xsdattgroup"]
*[class~="xml-d/xsdcomplextype"]
*[class~="xml-d/xsdgroup"]
*[class~="xml-d/xsdsimpletype"]
{
    font-family:monospace;
}
*[class~="xml-d/xmlelement"]:before {
    content: '<';
```

```
}
*[class~="xml-d/xmlelement"]:after {
    content: '>';
}

*[class~="xml-d/parameterentity"]:before {
    content: '%';
}

*[class~="xml-d/textentity"]:before {
    content: '&';
}

*[class~="xml-d/numcharref"]:before {
    content: '&#';
}

*[class~="xml-d/parameterentity"]:after,
*[class~="xml-d/textentity"]:after,
*[class~="xml-d/numcharref"]:after
{
    content: ';';
}

*[class~="xml-d/xmlatt"]:before {
    content: '@';
}
```

```
<topic id="mathml-test-topic-01">
  <title>XML Domain Test</title>
  <body>
    <dl>
      <dlentry>
        <dt>numcharref</dt>
        <dd>
          <p><numcharref>x0a</numcharref>.</p>
        </dd>
      </dlentry>
      <dlentry>
        <dt>parameterentity</dt>
        <dd>
          <p><parameterentity>p.content</parameterentity></p>
        </dd>
      </dlentry>
      <dlentry>
        <dt>rngpattern</dt>
        <dd><rngpattern>p.content</rngpattern></dd>
      </dlentry>
      <dlentry>
        <dt>textentity</dt>
        <dd>
          <p><textentity>hi-d-att</textentity></p>
        </dd>
      </dlentry>
      <dlentry>
        <dt>xmlatt</dt>
        <dd><xmlatt>class</xmlatt></dd>
      </dlentry>
      <dlentry>
        <dt>xmlelement</dt>
        <dd><xmlelement>foreign</xmlelement></dd>
      </dlentry>
      <dlentry>
```

```
        <dt>xmlnsname</dt>
        <dd><xmlnsname>http://www.w3.org/2000/svg</xmlnsname></dd>
      </dlentry>
      <dlentry>
        <dt>xsdattgroup</dt>
        <dd><xsdattgroup>p.attributes</xsdattgroup></dd>
      </dlentry>
      <dlentry>
        <dt>xsdcomplextype</dt>
        <dd><xsdcomplextype>p.class</xsdcomplextype></dd>
      </dlentry>
      <dlentry>
        <dt>xsdgroup</dt>
        <dd><xsdgroup>p.content</xsdgroup></dd>
      </dlentry>
      <dlentry>
        <dt>xsdsimpletype</dt>
        <dd><xsdsimpletype>type-tmtype-att.class</xsdsimpletype></dd>
      </dlentry>
    </dl>
  </body>
</topic>
```

Figure 4: Sample topic with XML mention markup.