

DITA 1.3 proposed feature #13035

Contents

DITA 1.3 proposed feature #13035.....	3
--	----------

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
- 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 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? TBD, but may be none.
 - 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 to add the "&#" leading characters: "
"

parment

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

```
<parment>p.content</parment> defines
the content model
for <xmlelem>p</xmlelem>.
```

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

rngpattern

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

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

textent

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

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

A typical rendition would be to add the leading "&" 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".

xmlelem

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

```
The <xmlelem>foreign</xmlelem> 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
<xmlelem>p</xmlelem> element.
```

xsdcomplextyp

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

```
The complex
type <xsdcomplextyp>state.class</
xsdcomplextyp> defines
the base content model for the
<xmlelem>state</xmlelem> 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
<xmlelem>text</xmlelem> element.
```

xsdsimpletype

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

```
The simple type <xsdsimpletype>type-
tmtyp-att.class</xsdsimpletype>
defines the allowed values for the
<xmlatt>tmtyp</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 |
parment |
rngpattern |
```

```

textent |
xmlatt |
xmlelem |
xmlnsname |
xsdattgroup |
xsdcomplexttype |
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 % parment
"parment"
>
<!ENTITY % rngpattern
"rngpattern"
>
<!ENTITY % textent
"textent"
>
<!ENTITY % xmlatt
"xmlatt"
>
<!ENTITY % xmlelem
"xmlelem"
>
<!ENTITY % xmlnsname
"xmlnsname"
>
<!ENTITY % xsdattgroup
"xsdattgroup"
>
<!ENTITY % xsdcomplexttype
"xsdcomplexttype"
>

```

```

<!ENTITY % xsdgroup
  "xsdgroup"
>
<!ENTITY % xsdsimpletype
  "xsdsimpletype"
>

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

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

```

```

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

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

```

```

<!ATTLIST xsdcomplextypes %xsdcomplextypes.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 parment %global-atts; class CDATA "+ topic/keyword xml-d/
parment " >
  <!ATTLIST rngpattern %global-atts; class CDATA "+ topic/keyword xml-d/
rngpattern " >
  <!ATTLIST textent %global-atts; class CDATA "+ topic/keyword xml-d/
textent " >
  <!ATTLIST xmlnsname %global-atts; class CDATA "+ topic/keyword xml-d/
xmlnsname " >
  <!ATTLIST xmlatt %global-atts; class CDATA "+ topic/keyword xml-d/
xmlatt " >
  <!ATTLIST xmlelem %global-atts; class CDATA "+ topic/keyword xml-d/
xmlelem " >
  <!ATTLIST xsdattgroup %global-atts; class CDATA "+ topic/keyword xml-d/
xsdattgroup " >

```

```

<!ATTLIST xsdcomplextypes %global-atts; class CDATA "+ topic/keyword xml-
d/xsdcomplextypes " >
<!ATTLIST xsdgroup %global-atts; class CDATA "+ topic/keyword xml-d/
xsdgroup " >
<!ATTLIST xsdsimpletypes %global-atts; class CDATA "+ topic/keyword xml-d/
xsdsimpletypes " >

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

```

Figure 1: DTD Syntax domain module declarations

svgDomainMod.xsd:

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.

svgDomainMod.rnc:

Figure 3: RelaxNG Compact domain module declarations

Examples

```

<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>parment</dt>
        <dd>
          <p><parment>p.content</parment></p>
        </dd>
      </dlentry>
      <dlentry>
        <dt>rngpattern</dt>
        <dd><rngpattern>p.content</rngpattern></dd>
      </dlentry>
      <dlentry>
        <dt>textent</dt>
        <dd>
          <p><textent>hi-d-att</textent></p>
        </dd>
      </dlentry>
      <dlentry>
        <dt>xmlatt</dt>
        <dd><xmlatt>class</xmlatt></dd>
      </dlentry>
      <dlentry>
        <dt>xmlelem</dt>
        <dd><xmlelem>foreign</xmlelem></dd>
      </dlentry>
    </dl>
  </body>
</topic>

```

```

</dentry>
<dentry>
  <dt>xmlnsname</dt>
  <dd><xmlnsname>http://www.w3.org/2000/svg</xmlnsname></dd>
</dentry>
<dentry>
  <dt>xsdattgroup</dt>
  <dd><xsdattgroup>p.attributes</xsdattgroup></dd>
</dentry>
<dentry>
  <dt>xsdcomplexttype</dt>
  <dd><xsdcomplexttype>p.class</xsdcomplexttype></dd>
</dentry>
<dentry>
  <dt>xsdgroup</dt>
  <dd><xsdgroup>p.content</xsdgroup></dd>
</dentry>
<dentry>
  <dt>xsdsimpletype</dt>
  <dd><xsdsimpletype>type-tmtime-att.class</xsdsimpletype></dd>
</dentry>
</dl>
</body>
</topic>

```

Figure 4: Sample topic with XML mention markup.