

DITA 1.3 proposed feature #13111

Contents

DITA 1.3 proposed feature #1311..... 3

DITA 1.3 proposed feature #13111

Official domain for MathML

Date and version information

Include the following information:

- Proposal Submitted: 1 October 2012
- Change History:
 - 7 Oct 2012:
 - Updated RNC declarations to correct issue with handling of no-namespace elements
 - Updated DTD declarations to correctly configure namespace prefix declarations
 - Corrected `doctypes/rng` to `doctypes/rnc` in the implementation materials.
- Champion: Eliot Kimber,
- Email discussion: <https://lists.oasis-open.org/archives/dita/201205/msg00013.html>

Original requirement

The MathML vocabulary is mature and well-established. It is now a formal part of both HTML5 and EPUB3. It is supported to one degree or another by the latest versions of most Web browsers. It is supported by the main commercial XSL-FO engines. Design Science provides a free, open-source JavaScript MathML library that enables MathML rendering in any Web browser that supports JavaScript.

MathML is used by many existing DITA communities, including learning content, textbooks, and technical documents that involve mathematics.

There are a number of DITA MathML integrations in use, including locally-created integrations and the DITA for Publishers project. While there is no particular difficulty in defining a MathML vocabulary module, there's also no reason for there to be multiple. Having a TC-provided MathML module would serve the community by removing the need for groups to implement their own or use a less-standard alternative.

Use cases

A MathML integration supports any context in which mathematics of any sort need to be presented, especially where the equations need to be rendered in a variety of contexts and in accessible ways (such as screen readers for digital delivery). Typical use cases include:

- Display and inline equations in learning content
- Formulas used in support of user tasks or concepts in technical documentation
- Math and science textbooks
- Scientific, mathematics, engineering, and mathematics (STEM) scholarly publishing.

Benefits

Address the following questions:

- Who benefits: DITA users who need to include MathML in their content
- Expected benefit: Easy use of MathML without the need to define a custom MathML vocabulary domain. Out-of-the-box support for MathML delivery by common DITA processing tools.
- Potential users: Difficult to quantify, but many learning publications require it (almost all math and science-related publications), technical documentation that involves equations.
- Degree of positive impact: Significant, as it makes MathML available to all DITA users without the need to define custom vocabulary modules.

Costs

Costs:

- Maintainers of the DTDs, XSDs, and RNGs: 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? One new topic for new element type `<mathml_container>`
 - How many existing topics will need to be edited? None. Only change to existing topics is generated content model descriptions.
 - Will the feature require substantial changes to the information architecture of the DITA specification? No architectural change.
- Vendors of tools: Tool vendors may decide to support the new module directly. The nature of this support will depend on the type of processor. Many DITA-aware editors and output processors already support MathML to one degree or another.
- 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, `mathmlDomain`, that defines the following element type:

- `<mathml_container>`
Specializes `topic/foreign`. Allows as content the `<math>` element from the MathML 3 vocabulary, `<data>`, or `<data-about>`, as a repeating OR group.
- Include the MathML 3 declarations. See <http://www.w3.org/TR/MathML3/>.

Processors that need to support MathML rendering have a number of options:

- For HTML:
 - Generate inline MathML within the HTML. This is supported in the latest versions of most common browsers at the time of writing.
 - Use the open-source MathJax JavaScript library to render MathML in any browser that supports JavaScript. See <http://www.mathjax.org>.
 - Generate images from the MathML using open-source or commercial tools. See http://www.w3.org/Math/Software/mathml_software_cat_components.html.
- For PDF:
 - The commercial XSL-FO engines Antenna House XSL Formatter and RenderX XEP both support rendering of MathML to PDF.
 - Generate EPS, SVG, or images using open-source or commercial tools

`mathmlDomain.ent`:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- =====
      DITA MathML Domain

      Defines a specialization of <foreign> that contains
      MathML markup.

      DITA 1.3

      Copyright (c) 2012 OASIS Open
```

```

===== -->

<!-- ===== -->
<!--           Formatting DOMAIN ENTITIES           -->
<!-- ===== -->

<!ENTITY % mathml-d-foreign
    "mathml_container
    "
>

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

<!-- ===== End DITA MathML Domain Entities ===== -->

```

mathmlDomain.mod:

```

<?xml version="1.0" encoding="utf-8"?>
<!-- =====
    DITA MathML Domain

    Defines a specialization of <foreign> that contains
    MathML markup.

    DITA 1.3

    Copyright (c) 2012 OASIS Open

    ===== -->

    <!ENTITY % mathml_container          "mathml_container" >

    <!ENTITY % MATHML.prefixed "INCLUDE">
    <!ENTITY % MATHML.prefix  "m">

    <!ENTITY % mathml3.dtd
        SYSTEM "mathml3/dtd/mathml3.dtd"
    >%mathml3.dtd;

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

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

    <!ENTITY % mathml_container.content
    "
        (%MATHML.pfx;math |
        %data; |
        %data-about;)*
    "
    >
    <!ENTITY % mathml_container.attributes
    "

```

```

%id-atts;
%localization-atts;
base
  CDATA
  #IMPLIED
%base-attribute-extensions;
outputclass
  CDATA
  #IMPLIED

"
>
<!ELEMENT mathml_container %mathml_container.content; >
<!ATTLIST mathml_container %mathml_container.attributes; >

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

<!ATTLIST mathml_container          %global-atts;  class CDATA "+ topic/
foreign mathml-d/mathml_container ">

<!-- ===== End MathML Domain ===== -->

```

Figure 1: DTD-Syntax Vocabulary Module

mathmlDomainMod.xsd:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:m="http://www.w3.org/1998/Math/MathML"
  elementFormDefault="qualified">

  <xs:import schemaLocation="mathml3/mathml3.xsd"
    namespace="http://www.w3.org/1998/Math/MathML"
  />

  <xs:group name="mathml_container">
    <xs:sequence>
      <xs:choice>
        <xs:element ref="mathml_container"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>

  <xs:group name="mathml_container.content">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="m:math"/>
      <xs:group ref="data.elements.incl" minOccurs="0"/>
    </xs:choice>
  </xs:group>

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

  <xs:complexType name="mathml_container.class" mixed="false">

```

```

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

  <xs:element name="mathml_container">
    <xs:annotation>
      <xs:documentation>
        The mathml_container (&lt;<keyword>mathml_container</keyword>&gt;)
        element
        contains zero or more MathML equations, along with optional
        &lt;<keyword>data</keyword>&gt;
        or &lt;<keyword>data-about</keyword>&gt; elements, which act as
        metadata for the
        equations.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType mixed="false">
      <xs:complexContent>
        <xs:extension base="mathml_container.class">
          <xs:attribute ref="class" default="+ topic/foreign mathml-d/
mathml_container "/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>

</xs:schema>

```

Figure 2: XSD-Syntax Vocabulary Module

mathmlDomainMod.rnc:

```

# =====
# MathML Domain Module
#
#   Defines a specialization of <foreign> that contains
#   MathML markup.
#
#   DITA 1.3
#
#   Copyright (c) 2012 OASIS Open
# =====

namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
namespace m = "http://www.w3.org/1998/Math/MathML"

# Define the domain values of this module

domains-atts-value |= "(topic mathml-d)"

# Define domain extension patterns

mathml-d-foreign =
  mathml_container.element

# Extend the patterns with the domain contribution
foreign |= mathml-d-foreign

# Define elements content and attributes

# LONG NAME: MathML Container

```

```

mathml_container.content =
(
  external "mathml3.rnc" |
  data |
  data-about
)*
mathml_container.attributes =
  univ-atts,
  attribute outputclass { text }?
mathml_container.element =

  ## The bold (<b>) element is used to apply bold highlighting to the
  content
  ##       of the element. Use this element only when there is not some
  other more proper element. For
  ##       example, for specific items such as GUI controls, use the
  <uicontrol> element. This
  ##       element is part of the DITA highlighting domain.
  ##       Category: Typographic elements
  element mathml_container {
    mathml_container.attlist,
    mathml_container.content
  }
mathml_container.attlist &= mathml_container.attributes

#
# End of module
#

```

Figure 3: RNC-Syntax Vocabulary Module

This RNC module is required in order to correctly configure the base MathML declarations for use with no-namespace DITA elements.

```

default namespace m = "http://www.w3.org/1998/Math/MathML"
namespace local = ""

# redefine the anyElement to exclude elements in no namespace, that is DITA
elements
include "mathml3/mathml3.rnc" {
  anyElement =
    element * - (m:* | local:*) {
      (attribute * { text }
      | text
      | anyElement)*
    }
}

```

Figure 4: mathml3.rnc

Examples

Provide examples of the proposed feature. Include an example for each of the use cases. Be sure to include edge cases, if known.

Equation inline within a paragraph:

```

<p>MathML Inline: <mathml_container>
  <m:math display='inline'>
    <m:semantics>
      <m:mrow>
        <m:msqrt>

```



```

        <m:mrow>
          <m:msup>
            <m:mi>a</m:mi>
            <m:mn>2</m:mn>
          </m:msup>
          <m:mo>+</m:mo><m:msup>
            <m:mi>b</m:mi>
            <m:mn>2</m:mn>
          </m:msup>
        </m:mrow>
      </m:msqrt>

    </m:mrow>
  </m:semantics>
</m:math>
</mathml_container></p>

```

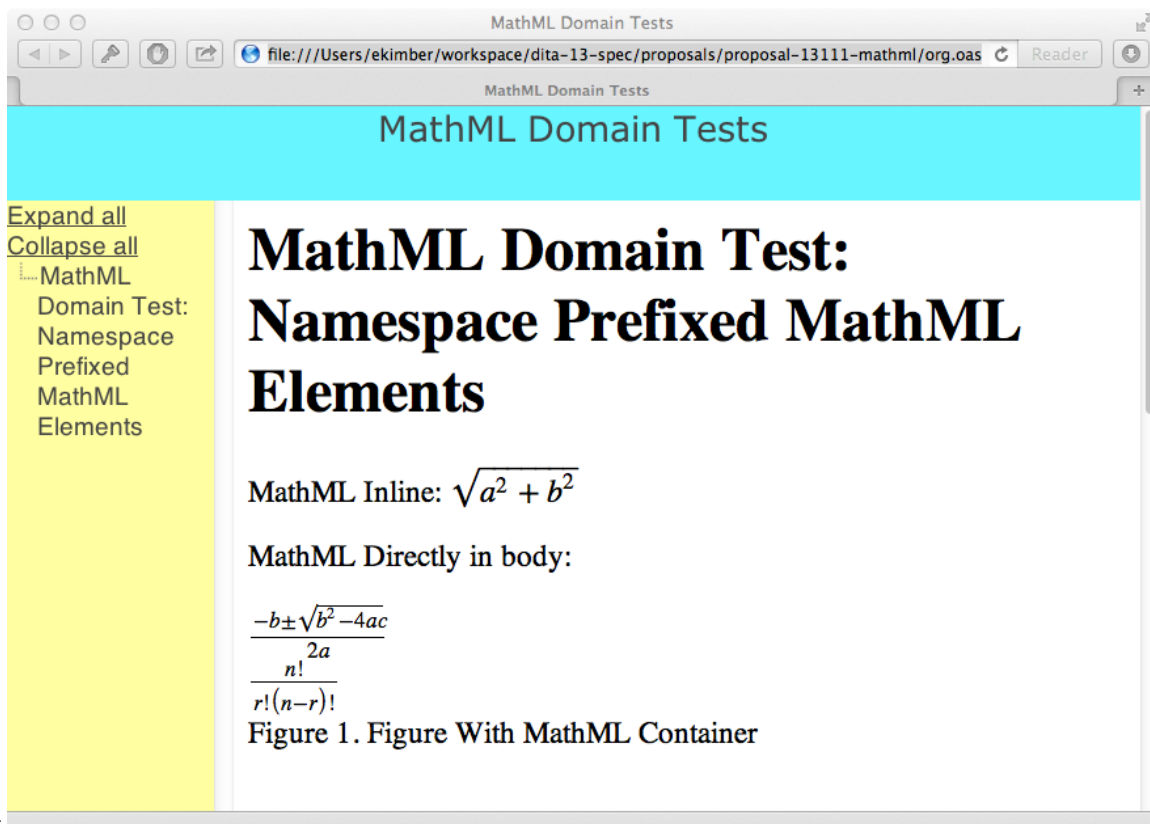
Display equation within a figure element:

```

<fig>
  <title>Figure With MathML Container</title>
  <mathml_container>
    <m:math display='block'>
      <m:semantics>
        <m:mrow>
          <m:mfrac>
            <m:mrow>
              <m:mi>n</m:mi><m:mo>!</m:mo>
            </m:mrow>
            <m:mrow>
              <m:mi>r</m:mi><m:mo>!</m:mo><m:mrow><m:mo>(</m:mo>
                <m:mrow>
                  <m:mi>n</m:mi><m:mo>&#x2212;</m:mo><m:mi>r</m:mi>
                </m:mrow>
                <m:mo>)</m:mo></m:mrow><m:mo>!</m:mo>
              </m:mrow>
            </m:mfrac>
          </m:mrow>
        </m:semantics>
      </m:math>
    </mathml_container>
  </fig>

```

MathML rendered to HTML using the DITA for Publishers math-to-HTML support.
The generated HTML uses the MathJax JavaScript plugin to render the math in the



browser.

Figure 5: MathML samples as rendered to HTML with MathJax for rendering