On the relationship between OpenMath and MathML

Bill Naylor Stephen Watt

Ontario Research Center for Computer Algebra
University of Western Ontario
London Ontario CANADA N6A 5B7

{bill,watt}@orcca.on.ca

July 6, 2001

Abstract

MathML provides a rather complete set of primitives for notational representation of mathematics but only a limited set of primitives to describe the meaning of mathematical expressions. OpenMath provides an extensible mechanism to describe mathematical semantics and is not concerned with notational nuances. Both MathML and OpenMath give XML representations for mathematical objects.

This paper describes how OpenMath may be used to extend MathML and *vice versa*: Two mechanisms are described whereby OpenMath content dictionaries may be used as an extension mechanism for MathML, overcoming the limitations in MathML's content model. Further, a description is given of a possible use of MathML elements within OpenMath objects.

1 Introduction

MathML [1] is an emerging standard from the World Wide Web Consortium, W3C [2]. Two of its main design goals are:

1) to provide a means of representing "publication quality" layout (*presentation MathML*) of mathematics on the internet,

2) to provide a means of representing the semantic meaning of mathematics (*content MathML*), again with a major concern that this should be transferable between applications on a single computer or over a network.

MathML is an XML application, and as such MathML data may readily be converted into other document styles using XSLT [4] style sheets. It has been designed primarily with the aim of representing mathematics to the level of *K-12*. Since mathematical notation consist of a small number of relative positioning primitives, it is fairly well covered by MathML. Problems arise, however, in the representation of the meaning of general mathematical objects. One of the limitations with MathML as it stands is that the meaning of an element *resides* in the name of the tag. Therefore, to represent mathematical concepts not pre-defined in MathML it is necessary to refer to some external definitions.

A resource which may be able to provide a *definition server* may exist in the form of OpenMath [5] *Content Dictionaries*. OpenMath is an XML application whose primary goal is to serve as a communication medium for the semantics of mathematical objects. One of its primary concerns is that it should be extensible, as it is a huge task to represent the objects of *all* of known mathematics, and indeed we may need to represent mathematics which is not yet defined! Extensibility is provided via content dictionaries which are XML repositories of mathematical information.

The OpenMath concept can provide a definition resource which may be used for extending MathML, whilst XSLT style-sheets enable a simple way of performing a translation between a document which contains *new* elements and a document which contains the required semantic elements containing pointers to *Symbols* in OpenMath Content Dictionaries [10]. In this paper we describe how this translation takes place and give an example of an original document together with the transformed document and the content dictionary it points to.

2 Using OpenMath to extend MathML

2.1 The technology available

The need to refer to externally defined concepts was anticipated by the authors of the MathML specification, and it was designed so that these mechanisms would be compatible with OpenMath. There are two seperate ways in which such an external source of mathematical knowledge may be accessed from within a MathML document, these are: via `semantics` elements and using the `definitionURL` attribute.

A "semantics" element has a number of children. The first child is the element which is having semantic information attached to it. The second and subsequent children must be either `annotation-xml` or `annotation` elements. In the first case it contains XML data which specifies the semantics of the first child, for example a section of OpenMath. In the second case it may be non-XML data for example a Maple command.

Example The MathML segment below attaches the meaning of the `BesselJ` symbol from the `specialFunctions` content dictionary to the element `<ci>J</ci>`.

```
<semantics>
  <ci>J</ci>
  <annotation-xml encoding="OpenMath">
    <OMS cd="specialFunctions" name="BesselJ"
         xmlns="http://www.openmath.org/OpenMath"/>
  </annotation-xml>
</semantics>
```

A "definitionURL" attribute may be used with the elements `csymbol`, `declare`, `semantics` and any operator element. We may set the definitionURL to point at a symbol in an OpenMath content dictionary using the element `csymbol`, in the following way:

If the *new* symbol we wish to define is denoted by `<mi>new</mi>` and the OpenMath symbol is defined at the URL `http://www.openmath.org/cdName#newSymbol` then we could use the following XML fragment:

```
<csymbol encoding="OpenMath"
         definitionURL="http://www.openmath.org/cdName#newSymbol">
  <mi>new</mi>
</csymbol>
```

Example The MathML element below could usefully be taken to have the same meaning as the `BesselJ` symbol in the `specialFunctions` content dictionary.

```
<csymbol encoding="OpenMath"
         definitionURL="http://www.openmath.org/specialFunctions#BesselJ">
  <mi>J</mi>
</csymbol>
```

An OpenMath Content Dictionary specifies meanings for a collection of names, such as the Bessel function J of the examples above. In an OpenMath fragment, each symbol appearing indicates both the name of the symbol and the content dictionary in which it occurs. That is, in order to specify an OpenMath symbol (this will be represented by an `OMS` element, with attributes set as follows) with name `symbol_name` from a content dictionary called `cd_name`, one could use the XML fragment:

```
<OMS name="symbol_name" cd="cd_name"/>
```

In the most common use of a symbol (to represent the application of a function to a number of arguments) this would be enclosed in an `OMA` element (standing for *OpenMath Application*), and the following children would be its arguments in a specified order. We use this to provide either an appropriate OpenMath XML fragment for a `semantics` annotation or URI for a `definitionURL` attribute.

3 Using semantics elements

Given a document containing some new symbol, we may apply an XSLT style-sheet to that document to give a document which has equivalent but now semantically meaningful `semantics` elements. The semantics elements could have an annotation part which referenced a symbol in an OpenMath content dictionary thus endowing the element with semantic meaning.

In the following example we replace a new element by the element `<m:semantics>`, as was done in [9]. Assuming that the required content dictionary (see Appendix [6]) were available this element would be semantically meaningful.

Example Suppose we were to be given the following section of *extended MathML*, which contains the element `<xm:BesselJ/>`, undefined in the standard MathML:

(we assume that `m` is the MathML namespace [7], `xsl` is the namespace for the XSLT elements, `xm` is a namespace which contains the names of the new elements and `om` is the namespace containing the names of the OpenMath elements).

```
<m:apply>
  <xm:BesselJ/><m:cn>1</m:cn><m:cn type="rational">1<m:sep/>2</m:cn>
</m:apply>
```

If we apply an XSLT style sheet containing the following rule, then we shall replace all applications of <xm:BesselJ/> with appropriate <m:semantics> markup.

```
<xsl:template match="m:apply[m:BesselJ]">
<m:apply>
  <m:semantics>
    <m:ci><m:mo>J</m:mo></m:ci>
    <m:annotation-xml encoding="OpenMath">
      <om:OMS cd="specialFunctions" name="BesselJ"/>
    </m:annotation-xml>
  </m:semantics>
  <xsl:apply-templates select="*[2]"/>
  <xsl:apply-templates select="*[3]"/>
</m:apply>
</xsl:template>}
```

These “templates” are effectively re-write rules which transform the input and, on application, may activate further re-write rules.

Assuming that there were templates available which copied through (verbatim) <m:cn> elements, then this would transform the preceding extended MathML into the semantically meaningful element (assuming that the required content dictionary see Appendix [6] was available):

```
<m:apply>
  <m:semantics>
    <m:ci><m:mo>J</m:mo></m:ci>
    <m:annotation-xml encoding="OpenMath">
      <om:OMS cd="specialFunctions" name="BesselJ"/>
    </m:annotation-xml>
  </m:semantics>
  <m:cn>1</m:cn><m:cn type="rational">1<m:sep/>2</m:cn>
</m:apply>
```

We may have intended different behavior, for example, that the presentation information should render as $J_1(\frac{1}{2})$. This is possible, and only requires that an alternative style-sheet is given, in this case:

```
<xsl:template match="m:apply[m:BesselJ]">
<m:apply>
  <m:semantics>
    <m:mrow>
      <m:msub><m:mi>J</m:mi><xsl:apply-templates select="*[2]"/></m:msub>
      <m:mfenced>
        <xsl:apply-templates select="*[3]"/>
```

```
      </m:mfenced>
    </m:mrow>
    <m:annotation-xml encoding="OpenMath">
      <om:OMS cd="specialFunctions" name="BesselJ"/>
    </m:annotation-xml>
  </m:semantics>
  <xsl:apply-templates select="*[2]"/>
  <xsl:apply-templates select="*[3]"/>
</m:apply>
</xsl:template>
```

4 Using definitionURL attributes

We continue with the example of the previous section:

```
<m:apply>
  <xm:BesselJ/><m:cn>1</m:cn><m:cn type="rational">1<m:sep/>2</m:cn>
</m:apply>
```

We add semantics using the `definitionURL` attribute in the manner discussed earlier. Valid semantically meaningful code for this example using this technique is:

```
<m:apply>
  <m:csymbol definitionURL="http://www.openmath.org/specialFunctions#BesselJ"/>
  <m:cn>1</m:cn><m:cn type="rational">1<m:sep/>2</m:cn>
</m:apply>
```

We could achieve this using the following style sheet:

```
<xsl:template match="m:apply[m:BesselJ]">
  <m:csymbol definitionURL="http://www.openmath.org/specialFunctions#BesselJ"/>
  <xsl:apply-templates select="*[2]"/>
  <xsl:apply-templates select="*[3]"/>
</xsl:template>
```

We see that given a library of XSLT style sheets together with some resource which allows the storage of mathematical knowledge, for example a library of OpenMath Content Dictionaries, extension and indeed personalisation (relative to a set of XSLT stylesheets) of MathML is indeed possible.

4.1 Macro-aware translation

One of the uses made of MathML shall be the generation of TeX. If the semantic extentions introducing new elements, e.g. `<mx:BesselJ>`, are expanded prior to TeX generation, then the result will potentially lose usefull information. It is not unlikely that an author shall have TeX definitions corresponding to these functions, e.g. `\BesselJ`. The translation between TeX and MathML, with an automated correspondence between TeX definitions and XSLT rules has been investigated in [9]. We investigate how information may be stored in order to allow this translation in our paper [8].

5 MathML in OpenMath

There are two places in which MathML elements should occur naturally in OpenMath Content Dictionaries.

First, we propose that the `<Description>` element be extended to contain two new subelements, `<CommonName>` and `<Notation>`. The contents of the notation element(s) would be expressed as MathML.

Secondly, we note that each definition element may have Commented Mathematical Properties (`<CMP>` elements). These commented mathematical properties are intended for human readers, not proof systems, and could be expressed naturally in MathML to facilitate browsing.[1]

Example Here we show MathML in use in the proposed `<Notation>` element.

```
<CD>
    <CDName>Bessel</CDName>
    <CDURL>http://www.openmath.org/CDs/bessel.ocd</CDURL>
    <CDReviewDate>2001-07-22</CDReviewDate>
    <CDStatus>experimental</CDStatus>
    <CDDate>2001-07-22</CDDate>
    <CDVersion>1</CDVersion>
    <CDRevision>0</CDRevision>
    <CDUses> </CDUses>

    <Description>
      This content dictionary contains symbols to describe
      the bessel functions and associated functions.
    </Description>
```

[1]Here we refer both to OpenMath `CDDefinition` elements and OMDoc `definition` elements

```
  <CDDefinition>
      <Name>BesselJ</Name>
      <Description>
         <CommonName>Bessel Functions of Integer Order</CommonName>

         <Notation>
             <version>
                <img src="besseleqn.jpg"/>

                <tex>J_\xref{arg1}{\nu}(\xref{arg2}{z})</tex>

                <math>
                   <mrow>
                     <msub>
                       <mi>J<mi>
                       <mi xref="arg1">&nu;</mi>
                     </msub>
                     <mo>&ApplyFunction;</mo>
                     <mfenced>
                        <mi xref="arg2">z</mi>
                     </mfenced>
                   </mrow>
                </math>
             </version>

             <FMP>
                <OMOBJ>
                    <OMA>
                       <OMS name="BesselJ" cd="bessel"/>
                       <OMV id="arg1" name="nu"/>
                       <OMV id="arg2" name="z"/>
                    </OMA>
                 </OMOBJ>
             </FMP>
         </Notation>

         The solutions to the differential equation ...
      </Description>

      ...

   </CDDefinition>

   ...

</CD>
```

These elements give various possibilities for how a symbol is intended to appear textually. Within a `Notation` element we shall allow

1) **version elements** There will be one of these for each notation used for this symbol. Each version element has some or all of the following children:

   i) **img element**, a URI specifying an image (for example a `gif` file) giving some sort of generic form of the notation (to be used in pull-down menus and the like).

   ii) **tex element**, This will contain a string which is easily translateable to TeX, this would be usefull for a tool which translates between extended MathML and Tex as suggested in section 4.1.

   iii) **math element**, these contain *presentation MathML*, this specifies the notation in a browser understandable fashion.

3) **FMP**, these will normally be an application of the symbol to its arguments, the arguments being cross-linked with the relevant arguments in the corresponding math element. This is intended to be a template which specifies the ordering of the arguments.

It should be possible for a computer algebra system to accurately comprehend *extended* presentation MathML, with no ambiguity assuming that the relevant Content Dictionaries were present and that the specified notational styles are followed.

We discuss the use of `<Notation>` elements in more detail in our paper [8].

6 Conclusion

We have shown how extended tag sets can be useful in MathML. Expressions including new elements may be rewritten using XSLT stylesheets to produce either parallel markup with OpenMath annotations in `<semantics>` elements or to produce `definitionURL` attributes in `<csymbol>` elements. This allows semantically meaningful expressions to use extended mathematical concepts as though they were in the base MathML definition.

We expect that writing expressions in this way shall be more readily adopted by human and machine authors of MathML than explicit, detailed `<semantics>` or `<csymbol>` markup. This may in fact be the only way that a significant proportion of MathML documents contain meaningful content in some years' time.

Secondly, MathML notations may be usefully incorporated in OpenMath Content Dictionaries in such a way as to allow OpenMath objects to be rendered in different notational variants using MathML.

References

[1] *Mathematical Markup Language (MathML) Version 2.0*, R.Aubrooks, S.Buswell, S.Dalmass, S.Devitt, A.Diaz, R.Hunter, B.Smith, N.Soiffer, R.Sutor and S.Watt, http://www.w3.org/TR/2001/REC-MathML2-20010221/.

[2] World Wide Web Consortium, http://www.w3.org.

[3] W3Cs Editor/Browser, Amaya:http://www.w3.org/Amaya.

[4] XSL Transformations (XSLT) Version 1.0, J.Clark, http://www.w3.org/TR/1999/REC-xslt-19991116/.

[5] OpenMath: http://www.openmath.org

[6] OMDoc: http://www.mathweb.org/omdoc/

[7] *Namespaces in XML*, T.Bray, D.Hollander, A.Laymen, http://www.w3.org/TR/1999/REC-xml-names-19990114/

[8] B.Naylor, S.Watt: Meta stylesheets for the conversion of Mathematical documents into multiple forms. To appear.

[9] Igor Rodionov: Tools for MathML, MSc thesis, U Western Ontario, 2001.

[10] Xuehong Li: XML and the Commnunication of Mathematical Objects, MSc Thesis, U Western Ontario, 1998.

A Appendix

In this Appendix we contain a html form of a content dictionary which contains the definition of the symbol BesselJ. The content dictionary has been written as an XML file and then translated into the html style using an XSLT style-sheet designed for the task. In order that the MathML parts of the content dictionary render satisfactorily a MathML *knowledgeable* browser eg. Amaya [3], must be used.

OpenMath Content Dictionary: bessel

Canonical URL:
: http://www.openmath.org/CDs/bessel.ocd

CD File:
: bessel.ocd

CD as XML Encoded OpenMath:
: bessel.omcd

Defines:
: BesselJ

Date:

Version:
: (Revision)

Review Date:

Status:
: private

This content dictionary contains symbols to describe the bessel functions and associated functions.

BesselJ

Common Name:
: Bessel Functions of Integer Order

Notation:
: $J_{\nu}(z)$
: $J_{\pm\nu}(z)$

Formal Specification of the Notational Style:

```
<OMOBJ>
  <OMS cd="bessel" name="BesselJ"/>
  <OMV name="nu" xref="arg1"/>
  <OMV name="x" xref="arg2"/>
</OMOBJ>
```

BesselJ *nu x*

Description:
: The solutions to the differential equation $J_{\pm \nu}(z)$ $z^2 \frac{d^2w}{dz^2}+z\frac{dw}{dz} + (z^2-\nu^2)w=0$ are the Bessel functions of the first kind. It has a branch cut along the negative real axis in the complex z plane. When $\nu=\pm n$, it is analytic in the entire complex z plane.

Reference:

M. Abramowitz and I. Stegun, Handbook of Mathematical Functions, [9.1.20], [9.1.21]

Function:

(integer, real) $\rightarrow$ real

$(\mathrm{Z}, \mathrm{R}) \rightarrow \mathrm{R}$

Function:

(integer, complex) $\rightarrow$ complex

$(\mathrm{Z}, \mathrm{C}) \rightarrow \mathrm{C}$

Function:

(symbolic, symbolic) $\rightarrow$ symbolic

$(\mathrm{S}, \mathrm{S}) \rightarrow \mathrm{S}$

Defining Commented Mathematical property (CMP):

$$(J_{\pm\nu}(z) = w) \Rightarrow z^2 \frac{\mathrm{d}^2 w}{\mathrm{d}z^2} + z \frac{\mathrm{d}w}{\mathrm{d}z} + (z - \nu) w = 0$$

Defining Formal Mathematical property (FMP):


```
<OMOBJ>
        <OMA>
          <OMS cd="relation1" name="eq"/>
          <OMA>
            <OMS cd="arith1" name="plus"/>
            <OMA>
              <OMS cd="arith1" name="times"/>
              <OMA>
                <OMS cd="arith1" name="power"/>
                <OMV name="z"/>
                <OMI>2</OMI>
              </OMA>
              <OMA>
                <OMA>
                  <OMS cd="calculus1" name="diff"/>
                  <OMBIND>
                    <OMS cd="fns1" name="lambda"/>
                    <OMBVAR>
                      <OMV name="z2"/>
                    </OMBVAR>
                    <OMA>
                      <OMA>
                        <OMS cd="calculus1" name="diff"/>
                        <OMBIND>
                          <OMS cd="fns1" name="lambda"/>
                          <OMBVAR>
                            <OMV name="z3"/>
                          </OMBVAR>
                          <OMA>
                            <OMS cd="bessel.ocd" name="BesselJ"/>
                            <OMV name="nu"/>
                            <OMV name="z3"/>
```

```
                        </OMA>
                      </OMBIND>
                    </OMA>
                    <OMV name="z2"/>
                  </OMA>
                </OMBIND>
              </OMA>
              <OMV name="z"/>
            </OMA>
          </OMA>
          <OMA>
            <OMS cd="arith1" name="times"/>
            <OMV name="z"/>
            <OMA>
              <OMA>
                <OMS cd="calculus1" name="diff"/>
                <OMBIND>
                  <OMS cd="fns1" name="lambda"/>
                  <OMBVAR>
                    <OMV name="z2"/>
                  </OMBVAR>
                  <OMA>
                    <OMA>
                      <OMS cd="bessel.ocd" name="BesselJ"/>
                      <OMV name="nu"/>
                      <OMV name="z2"/>
                    </OMA>
                  </OMA>
                </OMBIND>
              </OMA>
              <OMV name="z"/>
            </OMA>
          </OMA>
          <OMA>
            <OMS cd="arith1" name="times"/>
            <OMA>
              <OMS cd="arith1" name="minus"/>
              <OMV name="z"/>
              <OMV name="nu"/>
            </OMA>
            <OMV name="w"/>
          </OMA>
        </OMA>
        <OMS cd="alg1" name="zero"/>
      </OMA>
    </OMOBJ>
```


eq (plus (times (power (*z*, 2) , diff (lambda [*z2*] . (diff (lambda [*z3*] . (BesselJ (*nu*, *z3*))) (*z2*))) (*z*)) , times (*z*, diff (lambda [*z2*] . (BesselJ (*nu*, *z2*) ())) (*z*)) , times (minus (*z*, *nu*) , *w*)) , zero)

Commented Mathematical property (CMP):

$J_{\nu}(z)=\sum_{k=0}^{\infty} \frac{(-1)^k}{k!\Gamma(\nu+k+1)}(\frac{z}{2})^{\nu+2k}$, $|arg z| /lt \pi$, ascending series

$$J_\nu(z) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!\Gamma(\nu+k+1)} \left(\frac{z}{2}\right)^{\nu+2k}$$

Commented Mathematical property (CMP):

$J_n(z)=\frac{1}{\pi} \int_0^{\pi} cos(n \theta− zsin \theta) d \theta$, integral representation

$$J_n(z) = \frac{1}{\pi}\int_0^{\pi} \cos(n\theta - z\sin\theta)\,\mathrm{d}\theta$$

Commented Mathematical property (CMP):

$J_{−n}(z)=(−1)^nJ_n(z)$, symmetry

$$J_{-n}(z) = (-1)^n J_n(z)$$

Commented Mathematical property (CMP):

$J^{\prime}_0(z)=−J_1(z)$, derivative

$$\frac{\mathrm{d}}{\mathrm{d}z} J_0(z) = -J_1(z)$$

Signatures:

sts
