

NAG Ltd



OpenMath Project

Recent Developments in OpenMath

DAVID CARLISLE

Title Page

Contents

Home Page



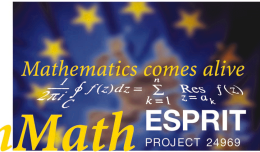
Go Back

Close

Quit

Contents

1 Introduction	3
2 CD Elements	4
3 Signature Files	5
4 Presentation Information	6
5 Inheritance	7
6 CDGroups	9
7 Formatting CDs	10
8 Core CD Distribution	11
9 OpenMath XML encoding	12
Grammar for XML Encoded OpenMath	13
UTF7 Encoding of strings	14
XML Comments	15
OpenMath Doctype, Namespace and stylesheets	16
10 OpenMath Compliance	17
Levels of OpenMath Compliance	18



OpenMath

OpenMath
David Carlisle

Title Page

Contents

◀◀ ▶▶

◀ ▶

Go Back

Close

Quit

Page 2 of 18

1. Introduction

A new draft of the OpenMath Standard is in development. In this talk I hope to highlight additional features that have been developed since the last OpenMath workshop.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 3 of 18

2. CD Elements

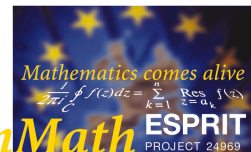
OpenMath Content Dictionaries are XML documents,.

They have certain elements which give “header” information about the CD itself, rather than about the symbols.

Existing elements include CDURL, giving the URL to the canonical copy of the CD, CDStatus, giving its status (public, experimental, etc).

It has been proposed that two new elements are added,

- CDDate The date the CD was last edited.
- CDVersion A version number (in the form major.minor).



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 4 of 18

3. Signature Files

OpenMath Objects may be typed with respect to more than one Type system.

To use a type system one first builds a CD giving the constructors for the type theory.

OpenMath Objects may then be built representing terms in the type theory, and the OpenMath Attribution mechanism may be used to assign types to OpenMath Objects.

In order that the Content Dictionary to be used with multiple type systems, the *Signature* element which was formerly used to express the signature of the symbol being defined has been removed in favour of separate *Signature Files*.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 5 of 18

4. Presentation Information

Previous releases of the draft standard have include in the CDs a Presentation element, which was intended to specify the presentation form of the symbol.

However it was never documented how this presentation specification should refer to the arguments of the symbol. This is required, for example to express the usual notation for integration, with limits being typeset as

$$\int_a^b$$

Current developments of the W3C Stylesheet language, XSL, mean that a better possibility might be to remove the Presentation element, and instead “associate” with each CD an XSL stylesheet that transforms OpenMath to Presentation MathML.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 6 of 18

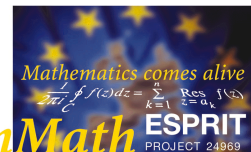
5. Inheritance

The previous OpenMath Draft suggested that future drafts would define a mechanism for specifying *Inheritance*.

A motivating example for inheritance would be a MathML Group.

The idea would be that a ‘MathML CD’ could be defined by some mechanism of including the individual CD that make up the areas of Mathematics covered by Content MathML.

This would allow an application to state simply that it supported the MathML CD, rather than having to list all the individual CD.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

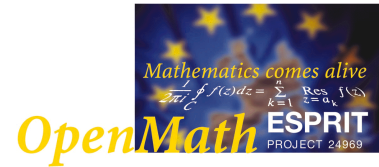
Quit

Page 7 of 18

The disadvantage of such a scheme is that it leads to incompatible objects. The symbol plus from the arith1 CD, would not be the same OpenMath object as the symbol ‘plus’ from the MathML CD.

In order that such objects behaved in the same manner in any OpenMath application, one would need to specify that all phrase-books would have access to the dependencies produced by the inheritance of CDs.

Such issues led us to instead propose the simpler scheme, of *CD Groups*.



OpenMath
David Carlisle



6. CDGroups

A CD Group is, as its name suggests, a name for a group of CDs.

The Group is specified by an XML document that gives a name for the group, and then lists the CDs that make up the group.

Other optional information in the CDGroup file includes Comments and header information similar to that present on CD files.



OpenMath

OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 9 of 18

7. Formatting CDs

The production of Content Dictionaries that correctly adhere to the format specified by the XML DTD, and the reading of these CDs requires the use of CD Tools.

Several tools have been developed within the Esprit project. In particular the **current draft set of Content Dictionaries** uses a set of XSL stylesheets.

These Stylesheets are used to present the Content Dictionaries in various formats, such as HTML, and also represented as XML encoded OpenMath, using the meta CD.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 10 of 18

8. Core CD Distribution



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

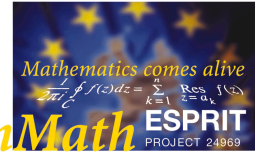
Quit

Page 11 of 18

9. OpenMath XML encoding

Apart from the CD related issues described above, the other main area in which outstanding issues remain is in the XML encoding of OpenMath.

A recurrent problem when using the XML encoding is to decide to what extent 'XML features' may be used in an XML document that is to be viewed as XML encoded OpenMath.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

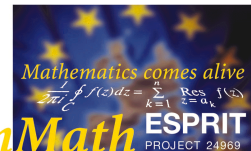
Quit

Page 12 of 18

Grammar for XML Encoded OpenMath

Currently the XML encoding is defined in terms of a grammar, some of whose productions are as follows:

S \longrightarrow (space|tab|cr|nl)+
object \longrightarrow symbol
| variable
| \langle OMI>S integer S \langle /OMI>
| \langle OMF S dec S? = S? "fpdec" S? \rangle /
| \langle OMF S hex S? = S? "fphex" S? \rangle /
| \langle OMSTR>S utf7 S \langle /OMSTR>
| \langle OMB>S base64 S \langle /OMB>
| \langle OMA>S? object S? objects S? \langle /OMA>
| \langle OMBIND>S? object S?
| \langle OMBVAR>S? variables S? \langle /OMBVAR>
| S? object S? \langle /OMBIND>
| \langle OME>S? symbol S? objects S? \langle /OME>
| \langle OMATTR>S? \langle OMATP>S? attrs S? \langle /OMATP>
| S? object S? \langle /OMATTR>



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 13 of 18

UTF7 Encoding of strings

The current requirement that OMSTR should take as content UTF7 encoded text is related to an early wish that the (as then SGML) encoding should be 'ASCII'.

As XML, and all XML tools, are inherently unicode based, it may be argued that this is no longer necessary. It also results in strings not being displayed legibly in XML tools such as editors, none of which currently support UTF7.

The current proposal is to remove this encoding and allow the XML text (which by default will be UTF8 encoded unicode) to be placed in OMSTR.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

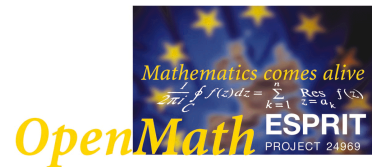
Page 14 of 18

XML Comments

The Grammar as written does not allow XML comments of the form

```
<!-- This is a comment -->
```

Comments may be easily added (and are in fact already supported by most OpenMath applications) but other XML constructs such as entities ‘&xxx;’ may be harder to support in OpenMath applications that are not full XML systems.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 15 of 18

OpenMath Doctype, Namespace and stylesheets

An XML document may declare that it uses the OpenMath DTD by starting with a declaration such as

```
<!DOCTYPE OMOBJ SYSTEM "omobj.dtd">
```

Of more importance to OpenMath, which one might typically want to embed in a larger document is the W3C Namespace mechanism. Using a Namespace attribute such as

```
<OMOBJ  
  xmlns="http://www.openmath.org/OpenMath">
```

tells the XML application that the OMOBJ element (and, by default, its children) are from the “OpenMath Namespace”.

Currently none of these constructs are allowed by the grammar.



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 16 of 18

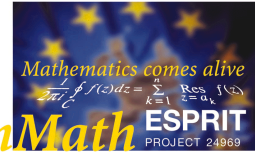
10. OpenMath Compliance

Recent drafts of the standard have *not* contained any section defining what it means for an application to be *OpenMath Compliant*. It was felt more experience was needed building OpenMath tools before such a specification could be given.

More experience has been gained now building demonstration OpenMath applications. It is proposed to update this section and reinstate it in the standard.

In the current working draft (June 1999) the *original* text from a early draft of the standard has been added. This text requires updating, but has been added to the current draft to allow discussion.

The section describes additional requirements at several levels, that can be viewed as representing increasing level of cooperation between applications:



OpenMath
David Carlisle

Title Page

Contents



Go Back

Close

Quit

Page 17 of 18

Levels of OpenMath Compliance

1. Additional rules an OpenMath application must follow regarding the OpenMath objects it can read and write. This is independent from any other application or form of communication. It is just some very basic requirements on how an application should treat symbols and content dictionaries.
2. In a context where an OpenMath application A can send an OpenMath object to another application B and B can send another object to A, constraints on what B can send in response to what A sent. At this level, we are mostly concerned about how common errors must be reported.
3. In a context where two OpenMath applications A and B have established a communication channel enabling them to exchange OpenMath objects, constraints on the sequence of OpenMath objects exchanged between them.
4. In the context of two applications A and B that want to communicate, methods to establish the communication.



OpenMath
David Carlisle

