Towards OPENMATH Version 2

MICHAEL KOHLHASE

School of Computer Science
Carnegie Mellon University
Pittsburgh, USA

Fachbereich Informatik
Universität des Saarlandes
Saarbrücken, Germany

http://www.cs.cmu.edu/˜kohlhase

(supported by the German Research Council under a Heisenberg grant)

OpenMath Version 2?

- The Pisa OpenMath meeting (Sept.. 2002) decided to prepare a new version of the OpenMath standard to
  - track Xml developments since 2000
  - take advantage of the practical experiences with OpenMath
- Committee: Stephen Buswell, Olga Caprotti, David Carlisle, Mike Dewar, Marc Gaetano, Michael Kohlhase.
  - Charter: prepare a new standard proposal (for Eindhoven?)
- Status: Discussions, but no coordinated proposal

Issues under Consideration for OpenMath 2

1. Basing OpenMath fully on XML (Status: Will do)
2. Structure of OpenMath objects
    1. (a) Structure Sharing for OpenMath objects (Status: DAG/Tree solution)
    2. (b) `OMDATA` for embedding XML data (Status: will do)
    3. (c) Namespaces/URIs for `OMS` (Status: ?????)
    4. (d) Types, `OMSuchThat`, first-class attribuitions (Status: ?????)
3. Extensions to the Content Dictionary format (Status: ????)
    1. (a) RDF encoding of CDs, allowing OMDoc
    2. (b) Defining a minimal data/functionality model
    3. (c) Conformance issues
4. Still open for suggestions (please contact us)

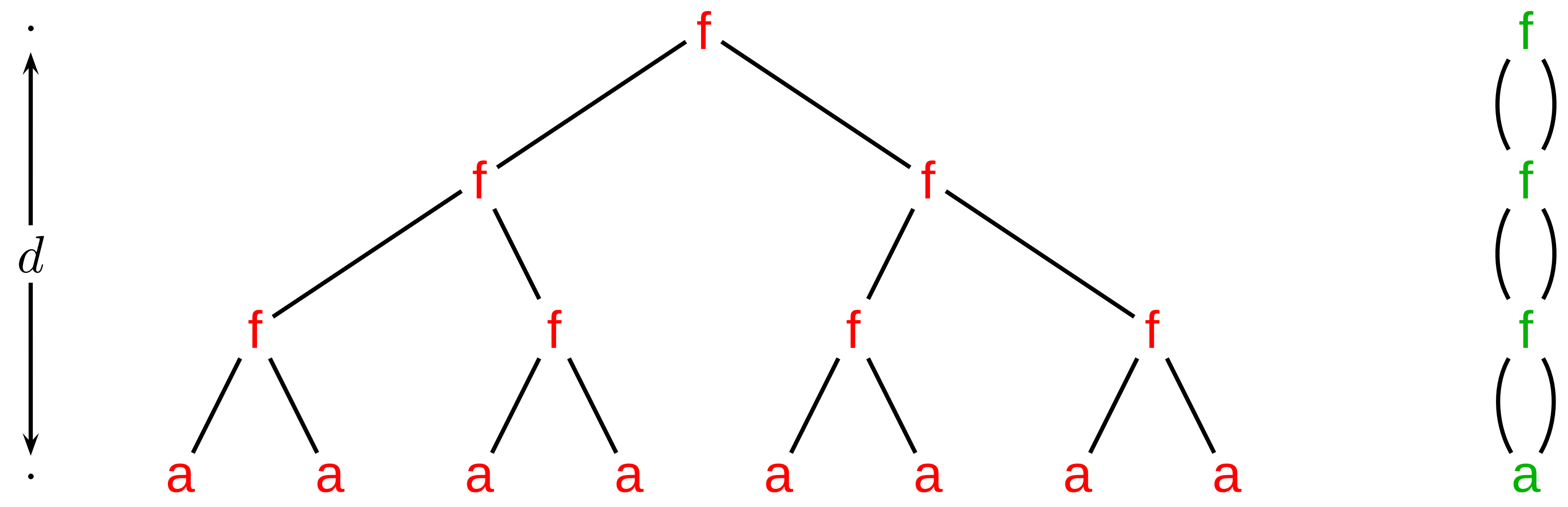
Issue: Basing the XML encoding fully on XML

- Background: OPENMATH 1 defines data model (trees) for OM objects
  - XML encoding: restricted subset based on byte-level grammar
  - binary encoding: byte-level serialization of OPENMATH object trees
- Decision: Allow arbitrary XML for the XML encoding
  - OPENMATH 2 is an XML application
  - get rid of byte-level grammar (was getting unwieldy anyway)
  - OPENMATH DTD and Schema become normative
  - could allow arbitrary UniCode in `cd` and `name` attributes (⚠ also affects the object model and binary encoding!)
  - XML goodies like entity references, namespaces, PI, everywhere

Structure Sharing for OpenMath objects

Tree

DAG



$2^d - 1$ nodes

d nodes

The same in the OpenMath XML encoding

```
<OMOBJ>
  <OMA>
    <OMV n="f"/>
    <OMA>
      <OMV n="f"/>
      <OMA>
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
      <OMA>
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
    </OMA>
    <OMA>
      <OMV n="f"/>
      <OMA>
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
      <OMA>
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
    </OMA>
  </OMA>
</OMOBJ>
```

```
<OMOBJ>
  <OMA>
    <OMV n="f"/>
    <OMA id="t1">
      <OMV n="f"/>
      <OMA id="t11">
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
      <OMR xlink:href="t11"/>


    </OMA>
    <OMR xlink:href="t1"/>









   </OMA>
  </OMA>
</OMOBJ>
```

Concrete Proposal

- Idea: Allow structure sharing in the XML encoding by
  - straw-man element `OMR` (represents target of `xlink:href` attribute)
  - by `id` attributes on OpenMath elements (possible targets)
- Pro: OpenMath data model does not change (stays finite trees.)

  Both encodings encode the OpenMath object

  application(f,**application**(f,**application**(f,a,a),
  application(f,a,a)),
  application(f,**application**(f,a,a),
  application(f,a,a)))
- Problem: Acyclicity Constraint (general DG represent infinite trees)

  non-local condition to be verified for validity
- Decision: Go for it, accompany with corresponding binary encoding

Issue: OMDATA for arbitrary XML data

- Problem: Want to allow XML data in attribuitions, e.g.

```
<OMBVAR>
  <OMATTR>
    <OMATP>
      <OMS cd="presentation" name="MathML"/>
      <OM???><msub><mi>X</mi><mn>4</mn></msub></OM???>
    </OMATTR>
    <OMV name="X4"/>
  </OMATTR>
</OMBVAR>
```

OMSTR and OMB are awkward for various reasons

- Decision: allow explicit OMDATA element in OMATP (attributes??)

Issue: Namespaces/URIs for OMS????

- Problem: Where to find the content dictionary, when we see

```
<OMS cd="foo" name="bar"/>?
```

- MathML solution: (use outright URI references)

```
<csymbol definitionURL="http://cds.foobag.org/foo.html#bar"/>
```

[+] web-conformance [–] location-independence/mirroring

- Namespaces solution: (use Namespaces)

```
<bar xmlns="http://cds.foobag.org/foo.html#bar"/>
```

[+] web-conformance [–] lose DTD validation [???] STS as schemas?

- URN solution: extend syntax, use Uniform resource names

```
<OMS cd="urn://cds.foobag.org/foo" name="bar"/>
```

[+] web conformance [–] need URN service

Issue: Types, OMSuchThat, first-class attribuitions

- Background: Attributions are second-class citizens of OpenMath
- OpenMath compliant applications need not even read them!
- types are represented using OMATTR, e.g.

```
<OMBVAR>
  <OMATTR>
    <OMATP>
      <OMS cd="sts" name="type"/>
      <OMA>
        <OMS name="mapsto" cd="sts"/>
        <OMV name="AbelianGroup"/>
        <OMV name="AbelianGroup"/>
      </OMA>
    </OMATTR>
    <OMV name="X4"/>
  </OMATTR>
</OMBVAR>
```

- types are second-class citizens of OpenMath

(Problem for formal methods people)

Solutions?

- First-class `OMATTR`: [+] General Solution [–] Difficult to control
- First-class types annotation: [+] solves Types [–] Special Solution

```
<OMBVAR>
  <OMTYPE cd="sts" name="type">
    <OMV name="AbelianGroup"/>
    <OMV name="X4"/>
  </OMATTR>
</OMBVAR>
```

- `OMBIND` with restrictions:
  [+] solves Types in logic, MathML compatibility [–] Special Solution

```
<OMBIND><OMS cd="quant2" name="rforall"/>
  <OMBVAR><OMV name="X4"/></OMBVAR>
  <OMSuchThat cd="sts" name="type">
    <OMA><OMS cd="sts" name="oftype"/><OMV name="X4"/><OMV name="AG"/></OMA>
  </OMSuchThat>
  ...
</OMBIND>
```

- any other ideas?

Issue: Extensions to the Content Dictionary format

- Proposal: RDF encoding of CDs (Buswell: Works, but what’s the use?)
- Proposal: allowing OMDoc (Overkill in many situations)
- Tentative Solution Defining a minimal data/functionality model
  - Model after OPENMATH objects: define data model/encodings
  - let 1000 flowers bloom
- Conformance issues (When is an application OPENMATH conformant)
  - can only be solved when CD data model is fixed
  - layers of conformance? syntactic, semantic, verified, types, . . .

Conclusions

- Discussion ongoing (please give us your input)
- Draft standard for next OpenMath Network meeting.
- Working Draft at `http://www.openmath.org/standard/om20`