Towards OpenMATH Version 2

Michael Kohlhase

(C) Michael Kohlhase

Supported by the German Research Council under a Heisenberg Grant

http://www.cs.cmu.edu/~kohlhase

Pittsburgh, USA

Saarbrücken, Germany

Universität des Saarlandes

School of Computer Science

Faculty of Informatics

Carnegie Mellon University

(191x26)
The Pisa OMDMEATH meeting (Sept. 2002) decided to prepare a new version of the OMDMEATH standard to track XML developments since 2000.

Committee: Stephen Buswell, Olga Caprotti, David Carlisle, Mike Dewar, Marc Gaetano, Michael Kohlhase.

Charter: Prepare a new standard proposal to take advantage of the practical experiences with OMDMEATH.

Status: Discussions, but no coordinated proposal.

(for Eindhoven?)
Issues under Consideration for OPMATH

1. Basing OPMATH fully on XML

2. Structure of OPMATH objects
   (a) Structure Sharing for OPMATH objects (Status: DAG/Tree solution)
   (b) OMDATA for embedding XML data
   (c) Namespaces/URIs for OMS

3. Extensions to the Content Dictionary format
   (a) RDF encoding of CDs, allowing OMDoc
   (b) Defining a minimal data/functionality model
   (c) Tyres, OMSCheetah, first-class attributions
   (d) Types, OMSuchthat
   (e) Conformance issues

4. Still open for suggestions
   (please contact us)

©: Michael Kohlhase

Carnegie Mellon
Issue: Basing the XML encoding fully on XML

Background:

- OPENMATH DTD and Schema become normative
- get rid of byte-level grammar
- OPENMATH 2 is an XML application

Decision:

- Allow arbitrary XML for the XML encoding
- binary encoding: byte-level serialization of OPENMATH object trees
- XML encoding: restricted subset based on byte-level grammar
- XML encoding: restricted data model (trees) for OM objects

Also affects the object model and binary encoding!

XML goodies like entity references, namespaces, PI, everywhere

-was getting unwieldy anyway

Decision: Allow arbitrary XML for the XML encoding

(c): Michael Kohlhase

4
Structure Sharing for OpenMath objects
The same in the OpenMath XML encoding
Concrete Proposal

Idea: Allow structure sharing in the XML encoding by – straw-man element OMR (represents target of *think:* here: attribute)

OMR (represents target of xlink: href) by id attributes on OPENMATH elements – by *think:* attributes on OPENMATH data model does not change

Problem: Acyclicity Constraint

(possibly targets)

Both encodings encode the OPENMATH object (stays finite trees)

Pro: OPENMATH data model does not change

Decision: Go for it, accompany with corresponding binary encoding

non-local condition to be verified for validity (general DG represent infinite trees)

application

application(f(application(f(a),a)),a)

application(f(application(f,a)),a)

application(f(application(f,a)),a)

application(f(application(f,a)),a)

application(f(application(f,a)),a)

application(f(application(f,a)),a)

©: Michael Kohlhase

Carnegie Mellon
Issue: 

Want to allow XML data in attributions, e.g.

\[ x^4 \]

OMSTR and OMB are awkward for various reasons.

Decision: allow explicit OMDATA element in OMATP.

(Attributes?)
Problem: Where to find the content dictionary, when we see `<OMScd="foo" name="bar"/>
</OMS>

**Issue: Namespaces/URIs for OMS???

**Namespaces/URIs for OMS???

**Problem**: Whereto find the content dictionary, when we see `<OMScd="foo" name="bar"/>
</OMS>`

**Solution: Namespaces**

- use right URI references

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

**Solution: URN**

- extend syntax, use Uniform Resource names

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

**Solution: Web-conformance**

- lose DTD validation

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

**Solution: MathML**

- uniformity

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

**Solution: Web-conformance**

- location independence/mirroring

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

**Solution: Namespaces**

- location independence/mirroring

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

**Solution: Web-conformance**

- uniformity

```xml
<OMS cd="urn:cds.foobag.org#foo name="bar"/>
```
Problem for formal methods people

Types are second-class citizens of OPENMATH

Types are represented using OMA. E.g.

\[ \text{OMVAR} \]
\[ \text{OMATTR} \]
\[ \text{OMA} \]
\[ \text{OMS} \]
\[ \text{OMAP} \]
\[ \text{OMATTR} \]
\[ \text{OMVAR} \]

Background: Attributions are second-class citizens of OPENMATH.
Solutions?

First-class

OMATTR: [+ GeneralSolution – Difficult to control

First-class types annotation: [+ solvesTypes – SpecialSolution

<OMBIND with restrictions: [+] solvesTypes in logic, MathML compatibility [–] Special Solution

any other ideas?

Michael Kohlhase

Carnegie Mellon
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)

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...

Let 1000 flowers bloom

Carnegie Mellon

Michael Kohlhase
Conclusions

Discussion ongoing (please give us your input)

Draft standard for next OpenMath Network meeting.

http://www.openmath.org/standard/om20

Carnegie Mellon

Michael Kohlhase