MMT URIs for OMDoc 1.6

Michael Kohlhase

September 22, 2008

Abstract

We propose a new URI scheme tailored to referencing semantic objects in the OMDoc 2 format and discuss implementation issues.

1 Introduction

Uniform Resource Identifiers (URI [BLFM98][1]) are sometimes called the "plumbing of the web", and indeed, the ability to identify and retrieve resources via a universal reference is one of the key innovations that made the world wide web possible. This is even more true for the "Semantic Web" [BLHL01], where relations between resources are made explicit. So it is no surprise that the foundations RDF [MM04] and OWL [MvH04] use URIs as identifiers of semantic objects.

The main utility of URIs comes from the the fact that they can be used to locate resources deployed by a large set of methods, ranging from static files on web servers to virtual resources that are created by web services upon requests: static files on web servers lower the entry barrier for authors participating in web content creation and web services allow for much of the richness of the data web environment.

In this note we propose a new URI scheme tailored to referencing semantic objects in the OMDoc 2 format and discuss implementation issues.

2 Reference in OMDoc 1

In OMDoc 1.2 [Koh06], we also make use of URIs for identifying objects, e.g. in the `for` attribute in `imports` elements. Implicitly we restricted the use of URIs in OMDoc 1.2 to URLs (Uniform Resource Locators; i.e. they actually locate a resource that can be retrieved); basically restricting ourselves to the `html` web, where document fragments are

[1] Actually — following general W3C practice — we use IRIs (Internationalized Resource Identifiers [DS05]) which extend the URI character set to almost all of Unicode throughout this note without explicitly saying so. IRIs us to use more sensible names in OMDoc. The only processing overhead involved in this is that IRIs must be URI-encoded before being sent over the wire.

note.tex 8059 2008-09-22 09:18:28Z kohlhase

identified by `xml:id` attributes and identified by fragment identifiers of the form `#foo`. In OMDoc 1.1 we even had special URI fragment identifiers the form `#byctx(cd,name)` that allowed to reference — i.e. identify; not necessarily locate — any OMDoc element that has a `name` attribute (basically only the `symbol`). Reference "by context" is attractive, since

- the uniqueness conditions on `name` attributes are less severe than those of `xml:id` attributes which must be document-unique, leading to more composable documents
- the referencing scheme is more semantic and thus less brittle against moving document fragments around.

But as "reference by context" this was slightly under-defined, of limited utility, and difficult to implement, this proposal was dropped in OMDoc 1.2.

3 Semantic Reference in OMDoc 2

For OMDoc 2, we are fundamentally rethinking how we want to reference semantic objects in OMDoc. In particular, in the MMT module system [RK08] — a new and improved version of the development graph model [MAH01] that underlies OMDoc 1.2 — we have extended the objects that have `name` attributes to all semantic objects, including theory morphisms. This leads to totally new application of semantic reference. We can identify OMDoc document fragments that are not explicitly represented in the OMDoc document (collection), but only virtually induced by the surface representation as part of the "flattened document (collection)". Note that flattening can lead to an exponential increase in size of the document, conversely MMT modularization can lead to extended theory reuse and to much more concise documents. Moreover, non-explicit versions of this technique have been used extensively in developments of mathematics like the Bourbaki collection [Bou68, Bou74]; see [Lau07] for a discussion.

The ability to reference the whole flattened document collection considerably increases the utility of semantic referencing in OMDoc, but also highlights the implementation problems. I claim that the the latter are aggravated by our attempts to get by with Uniform Resource *Locators* rather than fully embracing the the idea to use URIs to *identify resources* and leave the retrieval of the resources to another part of the information architecture.

4 A new Resource Identification Scheme or OMDoc

I propose a new URI scheme for identifying semantic objects in OMDoc 2, based on the MMT system, which will be a central organizing principle in OMDoc 2. The MMT system uses triples $\langle g, t, n\rangle$ for referencing, where g is the URI of a document, t is a theory path, and n is a complex name. These correspond to, but greatly extend the OpenMath [BCC+04]

referencing triple given by the `cdbase`[1], `cd`, and `name` attributes on `OMS` elements. [RK08] also suggests a scheme for encoding these as regular URIs, but the scheme has the serious drawback that the MMT triple cannot be parsed back from URIs. BTW, the URI encoding of OpenMath triples currently proposed in the MathML3 specification has the same problem. EdNote(1)

The new URI scheme for a MMT triple $\langle g, t, n\rangle$ is `mmt-`$\langle\!\langle g\rangle\!\rangle$`?cd=`$\langle\!\langle t\rangle\!\rangle$`;name=`$\langle\!\langle n\rangle\!\rangle$ for instance, the MMT URI for the symbol `plus` in the `arith1` content dictionary would be

```
mmt-http://cds.omdoc.org/omstd/arith1?cd=arith1;name=plus
```

which is still quite manageable to write by hand. The beauty of this approach which puts the semantic information into the query part of a URI is that we can add other specification items to it. Most importantly for us: version information, e.e.

```
mmt-http://cds.omdoc.org/omstd/arith1?cd=arith1;name=plus;revision=4711
```

Of course, we can have relative URIs as well: if g (the "cdbase") is known, then we just have `mmt:?cd=arith1;name=plus`. We might even go the extra mile and define `mmt:plus@arith1` as a synonym of the above.

5 Retrieval of MMT URIs

A great problem of using URIs is that it becomes totally unclear how to retrieve them. For some URI schemes, this can be solved by a simple cataloging service. In the OMDoc context this is much more difficult, since referencing the flattened collection requires non-trivial transformations. On the web services side, this functionality can be offered by the JOMDoc library [JOM] or — based on that — by the *OMBase* system [OMB]. But for a low entry barrier into OMDoc publication, we also want to accommodate for a way to deploy static OMDoc documents on simple web servers. On the other hand to achieve a low entry barrier for application developers mmt URIs should be easy to handle in applications like XSLT style sheets.

The main idea to achieve this is to externalize the flattening from the retrieval process which is relegated to regular retrieval methods. In the simplest case, we have a MMT web service MMTWS running at `http://mmt.omdoc.org/mmtws`. Then we can process the MMT URL above by rewriting it to

```
http://mmt.omdoc.org/mmtws?
   doc=http://cds.omdoc.org/omstd/arith1;cd=arith1;name=plus
```

by dropping the `mmt` prefix and moving the URI part before the query into the query[2]. EdNote(2)

[1] EdNote: There is still a mismatch here between the use of cdbase in OpenMath and the MMT usage. This should be taken care of before deploying.

[2] EdNote: I am not sure here, do we need to URI-encode the URI here? I think now

This URI the MMTWS processes by loading `http://cds.omdoc.org/omstd/arith1.omdoc`, flattening the theory `arith1` — possibly loading imported theories, and returning the element with name `plus` from the flattened theory as an OMDoc fragment[3]. The beauty of this setup is that EdNote(3)

- the rewriting step is the only think that needs to be done by the application developer. Any URI library or just raw string processing should this a relatively simple task.
- OMDoc authors can still deploy OMDoc documents or collections as static files on web servers without losing MMT modularity.

Thus it satisfies the "low entry barrier" requirement stated above.

6 Building MMTWS with the KWARC Toolkit

The MMTWS should be a relatively simple extension of the JOMDoc library once it can do theory flattening. Probably Florian already has a Scala implementation that is close to working for this, and he has been playing with the Scala web services toolkit I hear.

The MMTWS functionality could be enhanced in two ways in the future. The first idea is that we can just export a `mmtget` function from the JOMDoc library that can be directly called from say the `saxon` XSLT processor. This would probably be a good interface function for building MMTWS anyway.

The MMTWS web service could be based on *`OMBase`* instead of just JOMDoc, then it would have a caching facility and the web service would not have to load the relevant files over and over again. Assuming that most of the URIs used in the OMDoc documents are MMT URIs that are hosted by *`OMBase`* knowledge base services, most of the load operations will be svn update operations that only move diffs over the wire. I can see this as the beginning of a very effective and attractive *`OMBase`* cache network and I imagine always running a MMTWS/*`OMBase`* on my laptop, which would make me relatively independent of the network when I write OMDoc while traveling.

7 Caveats and Text Roadmap

This note is still in a very early stage, and is intended mainly as a vehicle or discussion between OMDoc developers. In particular, the URI scheme and all names are provisional and will probably evolve over time. At a later and more mature stage, part of the text might go into [RK08] and/or the OMDoc 1.6 specification.

[3] EdNote: we need to specify how to transport OMDoc fragments over the wire, so that they remain contextualized

8 Acknowledgements

This proposal has been greatly influenced by the HTTP Getter utility [Get] developed by Claudio Sacerdoti Coen and Stefano Zacchiroli for the HELM project [APCS01] at Bologna. Though I have been collaborating with them for years, I have never quite understood the relevance of the HTTP Getter until now.

References

[APCS01] Andrea Asperti, Luca Padovani, Claudio Sacerdoti Coen, and Irene Schena. HELM and the semantic math-web. In Richard. J. Boulton and Paul B. Jackson, editors, *Theorem Proving in Higher Order Logics: TPHOLs'01*, volume 2152 of *LNCS*, pages 59–74. Springer Verlag, 2001.

[BCC+04] Stephen Buswell, Olga Caprotti, David P. Carlisle, Michael C. Dewar, Marc Gaetano, and Michael Kohlhase. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004.

[BLFM98] Tim Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI), Generic Syntax. RFC 2717, Internet Engineering Task Force, 1998.

[BLHL01] Tim Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web - Computers navigating tomorrow's Web will understand more of what's going on making it more likely that you'll get what you really want. *Scientific American*, 284, 2001.

[Bou68] Nicolas Bourbaki. *Theory of Sets*. Elements of Mathematics. Springer Verlag, 1968.

[Bou74] Nicolas Bourbaki. *Algebra I*. Elements of Mathematics. Springer Verlag, 1974.

[DS05] M. Duerst and M. Suignard. Internationalized resource identifiers (IRIs). RFC 3987, Internet Engineering Task Force, 2005.

[Get] HTTP Getter Homepage [cited August 2008].

[JOM] JOMDoc Project [cited September 2008].

[Koh06] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006.

[Lau07] Bastian Laubner. Using theory graphs to map mathematics: A case study and a prototype. Master's thesis, Jacobs University, Bremen, August 2007.

[MAH01] Till Mossakowski, Serge Autexier, and Dieter Hutter. Extending development graphs with hiding. In H. Hußmann, editor, *Fundamental Approaches to Software Engineering (FASE 2001)*, number 2029 in LNCS, pages 269–284. Springer Verlag, 2001.

[MM04] Frank Manola and Eric Miller. RDF Primer. W3C Recommendation, World Wide Web Consortium, February 2004.

[MvH04] Deborah L. McGuinness and Frank van Harmelen. OWL web ontology language overview. W3C recommendation, W3C, February 2004.

[OMB] OMBase Project [online, cited April 2008].

[RK08] Florian Rabe and Michael Kohlhase. A web-scalable module system for mathematical theories. Manuscript, to be submitted to the Journal of Symbolic Computation, 2008.