

Twelfth OpenMath Workshop

Draft Notes of Eindhoven meeting 15/16.6.1999

22–26 people were present

These notes drafted by James Davenport
to whom corrections should be sent (jhd@maths.bath.ac.uk).

June 21, 1999

1. SB opened the meeting, and asked everyone to introduce themselves. AMC then spoke to local arrangements, and reminded people to pay the registration fee.
2. JHD spoke to the issue of content dictionaries. He reminded the meeting of the function of CDs. He introduced the `CDGroup` concept and the MathML `CDGroup`. He also introduced the concept of associated files, so that, to `transc1.ocd` (the 15 transcendental functions in MathML Content), there would correspond a `transc1.sts` for the Simple Type System, possibly a `transc1.ecc` for the Extended Calculus of Constructions, and hopefully a `transc1.xsl` containing an XSL stylesheet for rendering the objects in terms of MathML Presentation.

He discussed the immediate plans for CDs: revising `poly.ocd`, looking at the linear algebra CDs, writing a CD for dimensional analysis (and possibly one for units), and writing geometry CDs (n -dimensional with 2- and 3-dimensional extensions). In the longer term, OpenMath needed a CD server, both for humans and for software. More people also needed to be writing CDs. In discussion, people asked what tools were available: this was referred to after the next talk.

3. DPC spoke to the issue of the OpenMath standard. He commented that his talk could be sub-titled “changes since Tallahassee”. He introduced the XSL stylesheets which converted formal CDs to readable HTML, and to genuine OpenMath objects. He pointed out that an XML editor was very helpful in writing CDs: by hand it was very easy to miss out closing delimiters.

He mentioned ongoing changes: removal of some unused features of CDs, e.g. `signature`, and removal of the UTF-7 encoding, since UTF-7 was

an orphan of the encoding world. He also pointed out that the standard needed to address “compliance”: SD’s original text had been restored as a place-holder.

There was a discussion of the fact that not every XML conforming to the OpenMath DTD was actually valid OpenMath. For example, the DTD required the content of an `OMI` element to be `PCDATA`, but the OpenMath grammar insisted that the content be a (decimal) number. SB pointed out that MathML was in a similar position. There were many demands for a subset of XML, and we had hoped to use one of these — “XML profile”. However, this effort seemed to be going nowhere.

There was a request for a change log for the draft, and possibly making changes visible in the draft, e.g. change bars. **DPC**

4. AMC introduced a paper by himself and MK on “Defining Mathematical Properties”. He said that CDs did not necessarily introduce the logical meaning of mathematical symbols. OpenMath should involve the logic community more. While OpenMath has Formal Mathematical Properties (FMPs), there is no differentiation between definitions and consequences. Also, some objects do not have FMPs, e.g. subset. He suggested a new tag, `DefMP`, which would be like FMPs, but the `DefMP`s would have to define the mathematical object uniquely. At least in theory, the FMPs would then be formally proved as consequences of the `DefMP`s.

In the Esprit group, there were two objections: one that they would scare many potential users, and the other was that people might want different `DefMP`s. To the first, he answered that there were many features of OpenMath that not everyone used. For the second, he noted that signatures had been moved to separate files, and maybe this would be appropriate for `DefMP`s.

This led to a lively debate. GHG said that placing the `DefMP`s in separate files was a move against the general trend towards databases. MK in particular called for genuinely usable OpenMath tools, e.g. for Reduce and Maple. Many agreed with him.

5. MK spoke about MathWEB and the associated tools, such as the mathematical database MBase. The system uses KQML as the communication medium, and should use OpenMath as the content mechanism. The database should contain both formal information (definitions, assertions) and human-readable information. He claimed that the multiple files (see JHD’s talk) in OpenMath should really be considered as views from a central database. The primary objects in MBase were as follows.
 - (a) Symbols — rather similar to OpenMath.
 - (b) Definitions (if there are multiple definitions, then there has to be a proof that they are all equivalents).

- (c) Assertions. Some of these (Theorems etc.) will have proofs, others (Conjectures) will not.
- (d) Proofs.
- (e) Proof Objects, e.g. variants of a proof in different calculi, or in natural language. These can be stored using the “Propositions as Types” scheme.

It is important that the database is consistent, since an inconsistent database can generate anything. However, Gödel’s Theorem is a barrier here, so we try to ensure consistent extensions of a base set.

He stated that his project needed the following.

- (a) Clarification of the status of CDs and relation of these to a database.
- (b) Inheritance of CDs, e.g. we should learn from the Algebraic Specification project CASL.
- (c) Dynamic CDs to model communication.
- (d) Extension of OpenMath/MathML beyond K-12.

On the last point, SB asked for clarification. MK replied that there was currently no method of saying “let K be a field”, or that an `OMOBJ` is actually a theorem. DC questioned MK’s use of “parameterized” CD names. MK claimed that this was natural.

6. AS spoke about OpenMath, IAMC (Internet Accessible Mathematical Computation) and GAP. He therefore claimed that a standard protocol was needed for client-server communication. He outlined the following conundrum:

- Such a protocol is outside the scope of OpenMath.
- Such a protocol is essential for OpenMath.

He pointed out that GAP could be both a server and a client, e.g. both using Axiom and being used by Axiom. He saw three options.

- (a) KQML — see previous talk.
- (b) The Agent Communication Language from FIPA.
- (c) Wang’s MCP.

The last of these was illustrated. His `MCP.ocd` defined the concept of a “handle” as a way for the client to refer to objects on the server. His mechanism allowed for multiple content types. e.g. MathML as well OpenMath.

It was commented that the goals of AS were similar to those of CORBA and RMI. AS pointed out that some mechanism for clearing handles was

necessary. SB pointed out that what was described was not a complete protocol. GHG observed that these issues have been looked at before in the history of OpenMath. MCD had the report of the previous working party.

7. AMC opened the discussion session, saying that it was important to concentrate on the tabled draft of the OpenMath standard. He hoped that the meeting could endorse the draft on the next day. Chapter 6 (compliance) was very preliminary. He also observed that chapters 2–4 had been extensively discussed in Tallahassee. Chapter 5 had several amendments. He noted that there had been a decision to postpone any inheritance mechanism, in view of its difficulty, and those who had participated in this debate did not wish to see it re-opened.

AMC said that chapter 2 should mention signature files. DPC had hoped to add a section in Chapter 5 to enumerate these additional files. **DPC** MK asked what the mechanism was for defining new types of such files. MCD said that some files (e.g. `.sts`) would be standardised by OpenMath, but others, e.g. `.xsl`, would be in external formats, e.g. XSL. MCD thought that the syntax of signature files should be defined in the standard, but that it was not clear whether signature files should have the same “endorsed” standard as CDs.

AMC asked whether there would ever be an endorsed OpenMath phrase-book for, say, Maple? GHG thought not, but MK said that some official list should exist. BRM said that, whether or not they were endorsed, there should be a central repository of phrase-books.

AS asked whether CDs should not be OpenMath objects. DPC and SB pointed out that having the DTD for CDs made the use of XML editors feasible. It was pointed out that Chapter 2 should also mention CDGroup files. It was pointed out that no formal decision had been made on DefMPs (see above). **Workshop**

8. DPC said that Chapter 3 had only trivial changes.
9. On Chapter 4, GHG pointed out that back-references had disappeared from the XML encoding described in the document. DPC pointed out that XML itself did not support encoding. He said that there were existing XML linking mechanisms, and maybe these should be investigated. **DPC** AT asked whether white spaces should be allowed within the integers on OMI, to allow a large integer to be split across multiple lines. SMW pointed out that mathematical tables often used spaces to make numbers more readable. This suggestion was adopted. **DPC** to amend grammar.

Resumed on 16/6/1999.

10. TH spoke to the issue of translating OpenMath into C. The motivation was the use of a numerical library written in Fortran, which requires the user to produce sub-routines which can be called many times from the library routines. Since not all users will be familiar with Fortran/C, other means of specifying these subroutines are necessary. Many users may know Matlab or Maple. So maybe we should translate these into OpenMath and OpenMath into C. He reminded the meeting that there was an experimental programming CD in OpenMath v.1. There was also a `linalg.oed`. TH had written a Matlab CD, and also needed functions of the form `ApplyScalarToMatrix`. The code produced is very verbose, but an optimising compiler can produce quite good run-time behaviour: significantly faster than interpreted Matlab. However, debugging was not easy.

The OpenMath to C translator was comparatively simple, since TH used an off-the-shelf OpenMath parser. However, it was still necessary to consider issues such as memory management and name-space management. The bulk of the translator consists of the run-time library. TH said that people had noted that OpenMath is not a programming language. This is correct, but an algorithm is a valid mathematical object, and that is what is being writted in OpenMath.

AMC asked about comparisons with other efforts, e.g. MathLink. Th admitted that a direct translator would be more efficient, but the advantage of the OpenMath approach lay in its interoperability.

11. AT spoke about the use of OpenMath to communicate between CATHODE-II applications. CATHODE-II works on the solution of ODEs and PDEs, using a variety of computer algebra systems. They need a common demonstrator for the final report. CATHODE-II had several times considered OpenMath and rejected it as “vapourware”. Aldor had been adopted as a base for a file-based communication method. AH and Berth nevertheless wrote a CD for differential expressions.

In 1/1999 they had a demonstrator based on a heavily-modified version of the NAOMI JAVA libraries, and a significant amount of PERL ‘glue’. In 5/1999, they had a demonstrator which calls the ODEsolver in Reduce (the ZIB 1997 version) from Mathematica (version 3.0 with a package written Berth) *and returning the result*. He commented that MG’s input had been invaluable in finding the appropriate software. OpenMath is now adopted by CATHODE-II. OpenMath was not easy to use, and the various versions were out-of-date. It was hard to get up-to-date information.

Current tasks are to clean up this demonstration, and making it more general. We need a new OpenMath implementation for Reduce, possibly using the OpenMath ‘C’ library in the ‘C’ version of Reduce. However, Queen Mary tends to use the PSL version of Reduce, so some work neds to

be done. It would be possible to use MathLink in Mathematica, but in fact Berth's package is pretty good. They have re-written the NAOMI JAVA library to a more current version of OpenMath. "Aldor is an interesting problem": their version does not support Aldor. Bronstein might use (a subset of) the 'C' library linked to Aldor. He does not have a good solution to the Maple interface problem. It seems to AT that his group is doing a lot of generic work that, in some sense, OpenMath should be doing. In particular, there is no support from the official manufacturers. He also commented that if one goes to AltaVista, there is very little about OpenMath.

GHG apologized for the failure of Maple to communicate its support for OpenMath: there had been OpenMath support in Maple. AMC showed the new home pages for the OpenMath Society. AT had not seen these pages, and commented that they were essentially empty. AMC said that the management of the OpenMath pages would now be in the hands of the NAG group. MG pointed out that it was crucial for OpenMath to have commitments from algebra systems to provide at least β versions with OpenMath support. AT pointed out that he only had the NAOMI library, not the source code.

AT quoted the URLs

<http://cathode.maths.qmw.ac.uk> and

<http://paul.mat-inf.uni-greifswald.de/>.

12. AMC opened a continuation of the debate on the OpenMath standard. We have agreed to allow embedded white space. BRM thought that the DTD plus a few rules to constrain the PCDATA should be normative, rather than the grammar. Currently we can use off-the-shelf XML parsers, but not necessarily off-the-shelf generators. AMC said that he understood that there was a three-step process.

XML-verification — checking that the document conforms to the DTD.

OM-validation — checking that containers such as OMI contain what they should.

CD-validation — checking that symbols appeared in CDs; name-space attributes etc.

DPC said that there were two options: referring the grammar to the DTD, which implied that any OpenMath-compliant application had to be a full XML application; and having a separate grammar (which had the downside that BRM mentioned). The grammar was longer than it "should" be, since it had to cope with alternative orders of `name` and `cd` in OMS. If we were to allow more options here (e.g. namespace information) then the grammar would explode.

SB explained the position in detail: the grammar at 4.1 only generates a subset of XML, e.g. `<OMV name="x"/>` but not `<OMV name="x"></OMV>` or `<OMV name='x'/>`. On the other hand, the DTD can not enforce those parts of the grammar that correspond to the OM-validation stage above. He proposed that the grammar be reduced to perform only OM-validation, and that XML-validation should be defined precisely to be that defined by the DTD. JHD and others noted that this would probably imply removing the single quote from `varname` syntax, since `<OMV name="d'Alembert"/>` was valid XML, but `<OMV name='d'Alembert'/>` was not.

SD and MCD both thought that this a significant change, both editorially and in terms of the existing implementations: notably the INRIA libraries. DPC pointed out that there were half-way houses, e.g. allowing `name=` and `cd=` in either order, and allowing `'...'` as well as `"..."`. AMC asked whether we could write a tool that projected XML-documents onto the grammar of Figure 4.1. He proposed that we should adopt the DTD as the definition of XML-compliance. It was noted that at the moment some tools would only accept Figure 4.1. OM-validation would continue to be defined by (a descendant of) the relevant rules in the current 4.1. MCD pointed out that a proposal to change the standard should have a revised form of the text. It was agreed that a group of SB, SD DPC, BRM and AS, possibly with Steve Braham, would come forward with a proposal that would be discussed on the OM-list, but voted on by e-mail by the OpenMath Society.

Working Group

SMW asked why we allowed arbitrary large integers, but not floating-point numbers. JHD pointed out that page 10 stated that OM-floats were IEEE numbers. AS pointed out that this was an incompatibility with MathML, which allowed arbitrary-precision floating-point numbers. GHG proposed deleting the statement on page 10 about OMF not being big-float. JHD formalised this, by suggesting deleting the phrase “rather than bigfloats” on page 10, and changing “They”, to “When represented in hexadecimal, they”. This was agreed. It was noted that we needed to specify (or use IEEE to specify) the byte-order. BRM asked whether we should move to putting the data outside the attribute, c.f. `<OMI>`. SMW agreed. SD pleaded for stability in the encoding. If some-one wanted to propose this change, then they should produce an amendment, including a change to the binary encoding, which would be decided upon by the same mechanism as above. **AMC/OC** for the standard; **anyone** for the OMF change.

13. SB spoke to “Time, Truth and Type Theory”: this was sub-titled “Algorithms and Existence”. It responded to some questions raised in Helsinki. He said that a proof could be an OpenMath object, but there was no requirement to check that line i followed from its predecessors before checking line $i + 1$, as long as all lines were verified. This was not true of

algorithms. An algorithm was more similar to a conversation — see AS's talk. He felt that this was not quite enough, and proposed a “sequential execution operator” in addition. He then asked what the signature of `:=` was.

14. AMC led a further discussion on the standard, especially chapter 5. A few drafting errors were eliminated. It was clarified that `CDGroup` names could *not* be used to refer to symbols. It was agreed to list the content of the MathML `CDGroup` in an Appendix. In section 5.3, AS suggested that we should use emerging XML standards to encode bibliographic information in XML. DPC pointed out that this sort of cross-referencing was still under debate. It was agreed that issue should be returned too when the XML world had stabilised. DPC pointed out that 5.3.2 needed a certain amount of rewriting. AS asked whether there would be any way of storing the versions of CDs which were used in a particular archived object. MCD pointed out that there was no intent to invalidate stored objects, but that a new major version might invalidate software, e.g. by adding new symbols to a CD. MCD's proposal for major.minor version numbers was discussed at great length, but was finally accepted for `official` and `obsolete` CDs.

MK proposed that, in the light of the DefMP discussion above, which seemed to conclude that the DefMPs should be in auxiliary files, FMPs should be moved to a different kind of file. AMC agreed, but DPC did not. SB proposed, and JHD seconded, that FMPs should stay where they were. This was agreed. A few amendments to the DTD for CDs were noted. DPC pointed out that 5.4 (CD Signature files) and 5.5 (CD Groups) were probably not final. MK suggested a DTD for `defmp` files, which would be inserted after 5.4, after it had been discussed by e-mail. AS suggested that some tags like `CDVersion` should also be present in signature files.

To summarise, chapters 1–3 are agreed, chapter 4 is agreed modulo the XML-compliance issue, and some corrections, expansions and examples. JHD spoke to some changes to the CD world: adding FMPs to `set1.ocd`, writing `setname.ocd` (symbols `N`, `Z`, `Q`, `R`, `C`, `P`), which he was now using in `.sts` files, replacing `<OMV name="Integer">` by `<OMS name="Z" cd="setname">` etc.