Personal notes of RISC Summer School 25–30 June 2007

James H. Davenport
J.H.Davenport@bath.ac.uk

July 9, 2007

Contents

I OpenMath/JEM Joint Workshop 4

1 Monday 25 June 5

1.1 MathML roadmap . . . 5

1.1.1 General Changes . . . 5

1.1.2 Presentation Markup . . . 6

1.1.3 Content Markup . . . 6

1.1.4 The Compound Document Problem . . . 6

1.2 MathML Ontology . . . 6

1.2.1 To align or not to align? . . . 7

1.2.2 Conclusions . . . 7

1.3 Alignment Discussion . . . 7

1.4 OpenMath Content with Flexible Elisions — CL . . . 7

1.5 Content Dictionary Notations — PL . . . 8

1.6 A function reformulation of MathML-C2 — SD . . . 9

1.7 Visual Validation of MathML-P — PL . . . 9

1.8 MathDox — RV . . . 9

1.9 OMDoc Theory Morphisms— FR . . . 9

1.10 OpenMath: Symbols, CDs and Signatures — JHD . . . 10

1.11 OpenMath Society Meeting . . . 10

2 Tuesday 26 June 12

2.1 Why Powerpoint — MS . . . 12

2.2 MathML in Office — DPC . . . 12

2.3 Why TEX and LATEXML — BRM . . . 12

2.4 Markup for Mathematics — PI . . . 12

2.5 Access to Mathematics — BS/KM . . . 13

2.6 InkML and Mathematics — SMW . . . 14

2.7 JEM: Joining Educational Mathematics — OC . . . 14

2.8 SCIENCE: Scientific Computation Infrastructure in Europe — TK 15

II MathUI 16

3 Wednesday 27 June — Morning 17
3.1 Editing with MathType's TEX-syntax — RM 17
3.2 Authoring Interactive Exercies in ActiveMath — GG 17
3.3 A Semantic Wiki — CL 18
3.4 Wiris Conferencing for Mathematics — RE 18
3.5 Natural Editing of Algebraic Expressions — J-FN 18

III Calculemus 2007 20

4 Wednesday 27 June — Afternoon 21
4.1 Niche decision procedures — JH 21
4.1.1 Using types for verifying inductive invariants 21
4.1.2 Automated proofs of divisibility properties via ideal membership 21
4.1.3 Linear reasoning in normed spaces 22
4.2 Towards Constructive Homological Algebra — TC 22
4.3 What might "Understand a Function" mean — JHD 22
4.4 Biform theories in Chiron — WMF 23
4.5 Rational Reconstruction of a System for Experimental Mathematics — VS 23
4.6 Automatic Synthesis of Decision Procedures: A Case Study of Ground and Linear Arithmetic — PJ 24

IV MKM 2007 25

5 Thursday 28 June 26
5.1 Let's compute a proof: aspects of proving with computer algebra — PP 26
5.2 The Utility of OpenMath — JHD 27
5.3 Software Specification using Tabular Expressions and OMDoc — DP 27
5.4 First Steps on Using OpenMath to add proving capabilities to standard dynamic geometry systems — MA 28
5.5 Online Encyclopedia of Integer Sequences — NJAS 28

V Calculemus 2007 continued 30
5.6 Property Inference for Maple: An Application of Abstract Interpretation — JC 31
5.7 Towards practical reflection for formal mathematics — MG1 . . . 31
5.8 The efficiency of geometric theorem proving by means of Gröbner Bases — SM 32

5.9 Future Directions for Calculemus and MKM 32

6 Friday 29 June 35

6.1 A Collection of Elementary Problems in Geometry — TCH . . . 35

6.2 Rule-Based Simplification in Vector-Product Spaces — DJ 37

6.3 Quantifier Elimination for Approximate Factorization of Linear Partial Differential Operators — [SMC presented by WW] 37

6.4 Calculemus Business Meeting — VS 38

6.5 Mathematics and Scientific Notation — PM-R 38

VI Programming Languages for Mechanized Mathematics 40

6.6 Mei — A Module System for Mechanized Mathematics Systems — JX 41

6.7 Algebraic Structures in Axiom and Isabelle: a Comparison — CB 41

7 Saturday 30 June 43

7.1 PML — a new proof assistant — CR1 43

7.2 Declarative Representation of Proof Terms — CSC 44

7.3 Procedural Representation of CIC Proof Terms — FG 45

7.4 What Happened to Languages for Symboic Mathematical Computation — SMW 45

7.5 SML with antiquotations embedded into Isabelle/Isar — MW . . 46

7.6 Computer Algebra and the three 'E's:
Efficiency, Elegance and Expressiveness — JHD/JPff 47

7.7 PLMMS Business 47

A Dramatis Personæ 49

Disclaimer 1 *JHD's own notes. JHD has tried to capture some quotes, but these have not been checked by the quoted authors, so should not be attributed to them.*

Disclaimer 2 *JHD was on the OpenMath/JEM panel, so there are no notes of that. Also, at times he was trying to prepare his own talks, so the length of the notes is no function of the interest of the talk, not even subjective.*

Notation 1 *'-C' and '-P' refer to content and presentation markup for MathML.*

Part I

OpenMath/JEM Joint Workshop

Chapter 1

Monday 25 June

1.1 MathML roadmap

RM spoke to the MathML3 road map. In XML terms, MathML has always been, for better or for worse, an XML pioneer. If only we had had namespaces earlier, we wouldn't need to begin everything with `m`! MathML 2 (2nd ed.) was stable for years, with an interest group collecting questions and comments. Now chartered to produce MathML 3, with a chater March 2006–8. The WD has been released, which was "quite raw". `www-math@w3.org` is the public mailing list.

MathML 3 will be the main specification, with bits off-loaded, and cross-referenced.

MathML DOM will be a separate document. Much of this is automatically generated, and the MathML DOM as such is rarely used, since people tend to piggyback on the generic XML DOM.

XML Entity Names This has been a major, and very useful, piece of work, but is actually only loosely tied to MathML. This should therefore be a free-standing document, but still under W3C. HTML has some "legacy blunders" (JHD's summary), which we will probably have to live with.

MathML for CSS profile This will be a subset of MathML that can be rendered only using CSS, and therefore is a cross-platform rendering solution. Opera is a strong force behind this.

1.1.1 General Changes

- Condense and update historical and motivational material.
- Update references to Unicode etc.
- Replace `XLINK` with a MathML href attribute.

- MathML will maintain its ban on embedded markup from other namespaces. `semantics` and `maction` will allow some relaxation of this. There will also be a position paper on a comprehensive solution for scientific documents.

- `annotation` and `annotation-xml` will allow content by reference.

1.1.2 Presentation Markup

- a Math BiDi framework will support Arabic and other right-to-left languages.

- New concetps for elementary mathematics, such as long division, will be added. This is important for accessible MathML in Daisy. These changes should also allow for repeating decimals etc.

- Minor changes for `<mpadded>` will support high-resolution layout, of interest to publishers.

1.1.3 Content Markup

- Total rewrite.

- "canonical" versus "legacy", with canonical being much closer to OpenMath

- `<bind>` construct.

1.1.4 The Compound Document Problem

- The original Compound Document group of W3C has basically given up. (JHD's simplification)

- Equation numbers are a classic case of compound documents. Also support for assessment, diagrams etc.

1.2 MathML Ontology

MK spoke to this. MathML got all the glory while OpenMath "did the right thing". Alignment has been largely a function of DPC's stylesheets. Is it time for OpenMath 3? Content MathML is expression trees in prefix notation.

- MathML summations are more "natural" than the OpenMath ones, which are higher-order operators.

- MathML conditional sets are more "natural".

Where are the differences?

Symbols 100 fixed versus extensible.

Variables MathML variables import presentation, OpenMath ones are ‘frugal’. OpenMath variables should become more beautiful.

- *OpenMath only* has sharing, errors etc.
- *MathML only* has conditions.

1.2.1 To align or not to align?

Pro There’s only one standard: Microsoft etc. can’t say “which math”.

Con Changes of systems will be needed (on all sides).

Con Harder to improve the standards by playing the “but the other side ...” card.

Issue The MathML `condition`.

What is a CD — OpenMath, OMDoc, MathML?

1.2.2 Conclusions

We *should* align, to the point where MathML-C becomes an official encoding of OpenMath. This requires OpenMath 3, which should run to the same time-scale as (or possibly just lag behind) the MathML 3 standard.

1.3 Alignment Discussion

PI asked “how many technical difficulties are there”? MK thought the technical difficulties were soluble given sufficient qualified effort. RM asked if anyone thought alignment was not a “good thing”. No-one spoke out. JHD defined compatibility, and said that MathML had taken the courageous, right, decision, and OpenMath should do the same. JHD said that, as far as the OpenMath Society was concerned, CD review dates were not an obstacle, since CDs can always be marked as obsolete. A question from the floor asked whether one could use multiple CDs in the same document — yes.

1.4 OpenMath Content with Flexible Elisions — CL

The ‘expert’ tends to omit more brackets than the ‘novice’, but full explicit grouping makes things uneradable. This process is not limited to brackets. We therefore porpose a presentation mechanism that allows for flexible elisions. The model is tree (composition) full layout (elision) presentation. The ‘full layout’ would be `(a.x)+y`, elided to `ax+y` (with `&InvisibleTimes;`). Operators have

Fixity pre/post/mix.

precedence

associative full/left/right.

In theory XSLT is the answer, but in practice we need a short cut. OMDoc 1.2 is symbol-based, whereas Naylor–Watt is template-based. Level-1 templates are equivalent to symbols. `(log e x)` → `(ln x)` is an example that requires gerater depth. Isabelle models all symbol characteristics in one mixfix declaration. This is not sufficient for MathML-C and OpenMath, since n-ary operators are allowed. The following need to be treated:

- brackets due to precedence;
- default arguments $\log_{10} \rightarrow \log$;
- arguments can be inferred from others;
- arguments can be inferred from context.

The OpenMath must be annotated with some visiblity information, e.g. "compulsory" information that is needed to make sense of the expression in context.

1.5 Content Dictionary Notations — PL

We want to be able to render a symbol from a CD "found on the sidewalk". But this needs to respect user choices and cultures. We assume the rendered knows the user and context. As we have verb+.sts+ documents, so we should have `.ntn` documents, containing triples context/prototype/presentation, where prototype is an OMOBJ, and the presentation is in MathML-P. For example, `<OMS name="N" cd="setname1"/>` would render as $\mathbf{N}$ except in German, where it would be $\mathbf{N}^{+}$. This is implemented in ActiveMath, where there are linguistic renderings in English, Spanish, German and Russian. Another example is open intervals: German uses $]0, 1[$ for the angle-saxon $(0, 1)$.

He believes that this mechanism should be in both MathML 3 and OpenMath 3. The process for an author would be

1. identify a collect of .ntn files;
2. group them into a tree;
3. use meta-XSLT to produce an XSLT stylesheet;
4. the renderer uses this, along with the elision mechanism from the previous talk.

The current mechanism requires an extension to MathML-P to handle n-ary symbols better. He noted that pmath.xsl, being hand-authored XSL

1.6 A function reformulation of MathML-C2 — SD

- Improve alignment with OpenMath.
- Improve the structural extension possibilities of MathML-C.
- Improve MathML-C as a structural markup encoding.

MathML-P succeeds 'where there is general agreement on the structural conventions underlying the presentational forms'. SD referred to RM's corpus. OpenMath separates structure itself (the OpenMath standard) and semantics (the CDs). -C succeeds 'where there is general agreement on the semantics conventions underlying the functional structures'. Problems that need solving are special qualifiers like `logbase`, which would make MathML more 'functional'. Bound variables are a harder problem. Here MathML-C attempts to capture "common usgae', rather than some formal semantics as in `OMBIND`. Many such usages are similar to the 'iterator' in programming languages. There are probably equivalents of 'generator', e.g. in a VanderMonde matrix. SD believes that these two concepts will capture the MathML-C `bvar`.

"What about the namespaces"? OpenMath quotes a `cd=`, but this depends on the `cdbase`. We could get operator transparency with a uniform operator markup structure. MathML-C should be about the functional structure of an expression tree.

1.7 Visual Validation of MathML-P — PL

No full implementation, and there is some freedom for agents. **e-paper** is not a solution, since it has no respect of user wishes, and is not searchable.

1.8 MathDox — RV

Web pages are interactive, but don't support mathematics. MathDox is meant to answer this. The MathDox player transforms documents into interactive Web pages. It uses jsmath, and is used in ALT and ActiveMath.

1.9 OMDoc Theory Morphisms— FR

Generic specification of logics as theories in a logical framework. Interested in theory morphisms: T_1 is imported into T, written in L, whereas T'_1 is imported into T' written in L'. If we translate L into l', can we translate T into T'? Theories in OmDoc 1.2 can serve as OpenMath CDs. However, there are problems, as in the forbidden, but desired, double imports of monoid into ring. Needs a new constructor OMM to import symbols with renaming. However, this would require a concept of CD morphisms, possibly a good idea. Can do type-checking with respect to the semantics of the upper-most meta-language.

1.10 OpenMath: Symbols, CDs and Signatures — JHD

`http://staff.bath.ac.uk/masjhd/Slides/OMLinz-2007.pdf`. However, despite using `relation5` during the talk, the CD for total orders will be `total_order`: more mnemonic. In private conversation afterwards, DS pointed out that there's another, purely presentation-oriented, reason for not using '=' in the context $T(n) = O(n^2)$, and that is oral rendering. We would always "T of n is big O of n squared", and never "T of n equals big O of n squared".

1.11 OpenMath Society Meeting

MS opened the meeting, and welcomed 14 new members. He introduced a draft agenda, many of whose items were laid down by Finnish law/custom.

1. Election of Chair
2. Election of Secretary and Minute Checkers
3. Annual Report
4. Adoption of Balance Sheet and discharge of the Executive Committee
5. Election of the Executive Committee
6. MathML/OpenMath alignment
7. Any Other Business

This agenda was adopted. MS was elected Chair. OC was elected Minute Secretary. RM and MK were elected as Minute Checkers.

The Society had no formal activities in the last year, but the web site had been moved to DFKI (thanks to PL). There were no financial transactions. It should be noted that this was a good situation, since it meant that OpenMath is on track. PL said that more activity in the area of Content Dictionaries should be reflected on the web site.

MS said that there was no balance sheet, but this was corrected to read that there was a zero balance sheet. On this basis, the Executive Committee was discharged.

AMC had indicated a wish to stand down as President. The Meeting elected MK as President, MCD (continuing) as Vice-President, OC as Secretary of the Society, MG, MS and SMW as members-at-large. Although not an Executive Committee rôle, JHD was re-elected as Content Dictionary Editor.

The Chair of the meeting was handed to MK. He said that there seemed to be general agreement to produce a new OpenMath standard. He said that the next OM Society meeting should be in February/March 2008 (in order to meet the W3C timetable). JHD asked whether we *had* to produce OpenMath 3, or whether it might be more minor. MK thought that OpenMath 3 had good "marketing" rationale.

MK asked whether there was any other requirement for the new version. JHD raised the requirement for a "defining" property, which he would be in favour of. MK agreed with this. PL asked whether multiple such properties would be allowed. JHD said that that was for debate, but that they should certainly only be allowed if they were *proved* consistent.

MK proposed, and MS seconded, that we moved to a new OpenMath Standard, name to be decided. This was carried. Such a standard would need a working group to draft it. MK proposed himself, OC, JHD, DPC and MG. PL said that, if notations were to be included, he would like to be included. This group of six was formally constituted. JHD asked whether it was to be C_6 or S_3, since they were not isomorphic as Content Dictionaries. MK said C_6. The discussions of this group would be public by default.

After a question from PI, MK said that the group should also be commissioned to produce a set of Content Dictionaries aligned with MathML 3, to cover K–14. The goal should be a joint publication, W3C politics permitting.

The Meeting expressed its thanks to AMC for leading the Society, and MS for his able organisation.

Chapter 2

Tuesday 26 June

2.1 Why Powerpoint — MS

The real focus of this talk will be on distance and e-learning.

2.2 MathML in Office — DPC

Summary: there are bugs, but this is MathML working as was intended: expressions moving from IE to Word, being edited there, and then evaluated in Maple. In answer to a question, he stated that OpenOffice stores its mathematics in XML.

2.3 Why TEX and LATEXML — BRM

Why TEX using to be a trivial question: everyone did, and the typesetting was unsurpassed. But word processors are improving. The biggest bottleneck of DLMF is getting knowledge out of the heads of about forty mathematicians. For this, LATEX was obvious. But the D in DLMF implies the Web, hence LATEXML. This currently does LATEX-math to MathML-P: -C and OpenMath "someday". The program deals with TEX's "features", but generates XML rather than DVI.

MK reported that his team have run LATEXML over the Cornell e-print archive, with about a 60% success[1] ratio.

2.4 Markup for Mathematics — PI

In some sense, this talk is more about the history of notation. The Ischango bone is apparently a list of primes, dating to 5–10,000 BC. Bullae from Mesopotamia tabulate $N^3 + 3N^2 + N$ for $N = 1 \ldots 50$. Book "The Cossic Art" from 16th

[1] Defined as the process runs cleanly with no unprocessed macros.

century AD: ‘cos’ from Latin ‘causa’, meaning ‘the unknown’. Adam Ries (1492–1556) ‘Der Coß’. François Viète (1540–1603) used vowels for unknowns (probably the first to have multiple unknowns) and consonants for constants. Descartes (1596–1650) wrote ‘aa’ for ‘a times a’, but used integer superscripts for higher powers. Newton and Leibniz agreed (unusually! — JHD) in popularising the = sign, invented 100 years earlier. Euler produced $\int$, π and e. The 19th century saw determinants and matrices, and trees (Cayley), Maxwell’s equations (in Heaviside’s notation!), all of which were dependent on the evolution of the type-setters art.

Schröder spoke at ICM 1897 in Zürich (chaired by Peano) about the universal language of Pasigraphy, which we can now see as content markup. Pierce wrote (our notation) $e^{i\pi} + 1 = 0$ also as $\sqrt{e^{\pi}} = \sqrt[i]{i}$.

Computing introduced assignment, ++ and −−, Iverson introduced APL and its 88 symbols (because of the IBM Selectric golf ball). Page layout is now viewed as a form of programming. Markup used to be instructions to the typesetter, and was not for the reader (JHD: though it affected what the reader saw).

DS asked ‘who chose the font symbols’. PI quoted the origins of camels. PI said that there were symbols there that *he* had never seen, and whose etymology he had been unable to determine.

2.5 Access to Mathematics — BS/KM

KM began: “Math: the Mother of Science but the Bane of the Blind” — A. Karshmer.

Claim: the *authors* are responsible for accessibility, the ‘technicians’ can merely provide the tools. BS uses a Braille display to access mathematics. KM said that there were four fundamental problems:

Access to mathematical content;

Navigation of mathematical expression;

Presentation/Communication/Co-operation of mathematical information;

Doing mathematics.

A speech synthesiser is a *sequential* representation, as typically are Braille displays. Hence Linearity is fundamental. He quoted eleven systems for linear access, including their own LaBraDoor (ouch!).

But linearity is poor for navigating and browsing. Hence we need expand/collapse methodologies. Hence this really needds some content markup.

In the other direction, we should note that Braille notations are context sensitive, which makes translation for the sighted non-trivial. UCML is a library of Braille translations, e.g. U.S. Braille to German Braille.

Much ‘doing’ of mathematics is based on visual reasoning, e.g. re-arranging an equation. Therefore computer algebra functionalities may be more important for blind mathematicians.

BS said that, from his experience, the 'doing' was by far the hardest. As a teacher, devising problems for his students was very time-consuming: far more so than for his sighted colleagues. He illustrated by working $(3a + 5b - 4c) * (-5b + 4c - 6a)$, where his tool arranged the sub-computations and collected the results. Simplifying the collected result is equally demanding, and the tool supports this.

BS said that two-dimensional facilities for blind people was still very poor, partly because of haptic effects. MK added that designers do not seem to understand the real problems. CR added that the manipulation interface demonstrated would be useful for others as well.

2.6 InkML and Mathematics — SMW

"The pen is mightier than Word".

He would like to see 'power-assisted hand manipulation' Collaboration was also very important, be it in a teacher/class setting or distance collaboration. Most available tools for collaboration are vendor-specific and mathematically ignorant. Mathematical collaborations are between autonomous individuals, so heterogeneity is a given.

InkML is under the MultiModal Interaction (MMI) WG of W3C, which was chartered in February 2002. Microsoft was sceptical, quoting verbosity as a major problem!?[2], but this has essentially been solved[3]. InkML is intended to support applications from 'streaming ink' to 'archival ink'. Should support, say, pen and vioce synchronisation. The latest version has `annotation` and `annotationXML` (wonder where they came from). Dictionary-based methods are very powerful in ordinary hand-writing recognition, but essentially unavailable in mathematics. Therefore they have 'slurped' a large amount of data, and built a corpus of 5-grams, which shows, for example, that it is much more likely to be $\sin \omega t$ than $\sin wt$.

2.7 JEM: Joining Educational Mathematics — OC

`www.jem-thematic.net` is the web site. She spoke to the deliverables required, and the methodology of delivering them via the web site. Some-one has to take responsibility for the next set of deliverables, which are due on August 1st. The next workshop will be 21–23 February 2008 in Barcelona. The one after is planned for Trondhiem in Sept. 2008, but this has to be cleared with the EU. The first review will be in September 2007, but no details are known yet. **Members should communicate diary clashes to OC.** There are four new members of the network.

[2] JHD and SMW subsequently had a discussion as to whether the correct symbol here should be !? or ?!.

[3] SMW showed 'hello' in InkML, but adding 'world' would not fit on the screen.

MS also spoke, drawing attention to the need for all partners to engage fully with the project.

2.8 SCIENCE: Scientific Computation Infrastructure in Europe — TK

[Talk prepared by Alexander Konovalov.] This is a Framework VI infrastructure initiative programme. Main systems involved are GAP, KANT/KASH, Maple and Mupad. There are four networking activities, as well as transntaional access to facilities at RISC. All messages are represented in OpenMath, using the new CD `cascall1`. It is essentially a remote procedure call mechanism (JHD's understanding). The project wants to develop resource-brokering mechanisms that will support the irregular problem structure normally found in symbolic computation. Project URL: `www.risc.uni-linz.ac.at/projects/science/access`.

Part II

MathUI

Chapter 3

Wednesday 27 June — Morning

3.1 Editing with MathType's TEX-syntax — RM

He was demonstrating a β-version of MathType. MathType has always been a palette-based system — "point and click". MathType 6.0 will allow keyboard input in a "TEX-like" syntax, therefore allowing cut/paste and drag/drop. One observes that there are many "TEX-like" formats which are not actually TEX. The goal was Wikipedia plus "what came easily".

So we take what it is Texvc as documented in Wikipedia (plus information obtained by reverse-engineering the OCaml code), plus the MathML generated by Texvc when "it's simple enough" (i.e. to bypass image generation). `$` triggers TEX-mode, but β-testing led to the conclusion that, once users know there is TEX somewhere, they assume `^` and `_` have the usual meaning everywhere.

Wikipedia and Planet Math seem to work, but there are problems with MathWorld. However, "it's not that the bear dances well, it's that the bear dances".

3.2 Authoring Interactive Exercies in ActiveMath — GG

An exercise is more than pieces of mathematics, there are transitions, hints (a special kind of transition) and updating the learning model, a key element of ActiveMath. The exercise author can write questions/answers in terms of variables, and attach them to randomisers, so that the exercise is instantiated afresh each time. So far the randomiser only works over integers, but could choose, e.g. out of sets of functions.

During discussion, questions were raised, especially by CR, about "bad values" for random choices.

3.3 A Semantic Wiki — CL

SWiM — Semantic Wiki for Mathematical Knowledge Management. CL wanted to use a Wiki for mathematics, and found that the Wiki had to be semantic. The mathematics is represented in OMDoc. Existing items include Wikipedita, PlanetMath and Connexions. In the first two, searching for a formula is essentially impossible. In theory, this is possible in connexions, since MathML-C is used, but there is no current search facility.

> Show me all theorems about triangles which have a proof.

So how can OMDoc be integrated into a semantic Wiki? This poses the question, who is willing to author OMDoc/OpenMath? We must give the author added value as he/she is writing, such as 'instant gratification' by link suggestion — displaying the neighbourhood of the knowledge graph as it grows.

On a Wiki, one page represents one concept, which in SWiM is interpreted as being one statement or one theorem. CL therefore formalised OMDoc's three layers in OWL-DL. This might have wider implications for OMDoc. The overall methodology is extensible, and *should* be extended, to other sciences.

One could produce editing tools such as ontology-based auto-completion.

3.4 Wiris Conferencing for Mathematics — RE

Wiris CAS is an on-line CAS for mathematical education. This development is based on simultaneous access by two or more users to the same Wiris CAS session.

This produced was developed as a result of commercial pull from a large education company, for which mathematics is about 40% of their business. It started as an equation editor, but ... calculations and graphics soon followed.

Testing has just started: so far 16 users and 40 'conferences'. No major complaints so far! Speed is acceptable, but could be better. Need to store more in the XML, e.g. highlighting, graphical transformation.

3.5 Natural Editing of Algebraic Expressions — J-FN

'Natural' by analogy with "natural language". He was presenting work in the context of APLUSIX[1] – a system to help students to learn algebra. The student makes calculations, and the system verifies them. The student can use drag/drop to move terms from one side of an equation to the other. Operations like expand/simplify, move from numerator on left to denominator on right, clearing internal parentheses etc. are also available.

The teacher can decide what functions of the system, e.g. solve, are available at any point. The domain is numbers (integers, fractions, decimals and square

[1] `www.aplusix.com`.

roots), factorizations (one variable, degree ≤ 4; two variables, degree ≤ 2), equations and inequations, systems of linear equations. There are both training (with feedback) and testing modes, as well as features such as class statistics for the teacher, and various admistration features (enrolment etc.).

They tested the hypothesis “can students working alone with APLUSIX improve their capabilities” with 1000 grade 9 students in Brazil. Average correct answers/hour rose from 5.33 to 7.02. An experiment in India showed that collective learning seemed to work, and was received with pleasure (post-evaluation). APLUSIX does not ‘mark’ the work, so students can concentrate on the errors (as seen by student responses in post-evaluation).

Some teachers find the “trouble-shooter” rather than pedagogue rôle strange at first. APLUSIX has gone the commerical route because “that’s what publishers are there for”.

In the context of ReMap, they explicitly work with trees, so that the student can visualise the expression this way. There are exercises in asking the student to produce the tree format.

They are currently working on a ‘companion’ which can make suggestions and perform (and explain!) steps. Some graphical facilities need to be added. A facility for human annotation is needed. Also curriculum coverage must be increased to amortize the learning curve (more for staff than students). Curriculum has to be country- amd level-dependent: he sees this as being a greater obstacle than country-specific mathematical variants.

PL asked whether there might not be ‘LISP tutor’ syndrome[2]. J-FN said that he thought not, since APLUSIX provided less scallolding. Also the India evaluations has involved paper-based pre- and post-evaluations. ES asked about ‘dangerous’ simplifications such as $\frac{x^2-1}{x-1} = 0 \rightarrow x-1 = 0$. This was not regarded as a valid simplification by APLUSIX.

Coda

MathUI continued, but JHD switched to Calculemus.

[2] Apparently students used to the tutor can’t work once the scaffolding is removed.

Part III

Calculemus 2007

Chapter 4

Wednesday 27 June — Afternoon

4.1 Niche decision procedures — JH

Real applications throw up requirements for customised niche decision procedures. We may use refined features of the problem that general methods ignore. However, it often pays to solve a (slightly) more general problem than the one presented.

4.1.1 Using types for verifying inductive invariants

This commonly occurs with parametrized systems, e.g. N-fold Cartesian products. For simplicity here, we consider only full Σ_N symmetry. Example, N processors each with their own caching agents. Even if we assume that the individual processors are small and finite, we can still only use model checking for *given*, and often small[1], N. While we would like to use induction, simple proofs do not work, and one tends to need a much more complex induction invariant.

In principle, can use Bernays/Schönfinkel/Ramsey, so if there are no function symbols, the system is (large but) finite. In we use many-sorted Skolem/Gödel/ /Herbrand, then `Cache` has type `Node`→`State`, so `Cache(Cache(i))` is not well-typed, and the problem *does* become finite.

4.1.2 Automated proofs of divisibility properties via ideal membership

Proving that a formula is true for all integers can be done if it true for all reals, possibly with $x < y \Rightarrow x \leq y-1$. However, there are problems with $2x = 2y-1$.

So let's eliminate divisibilty in favour of existentials, e.g. "s, t coprime" $\Rightarrow \exists x, y sx+ty = 1$. This class of problems includes Hilbert's 10th. So generalise

[1] Intel have one for which $N = 2$ is doable, but $N = 3$ is already stressing the checker.

this to ideal membership in rings, and many practical problems can be solved here, e.g. by Gröbner bases (in theory over **Z**, but **Q** will generally do). An implementation in HOL-light is quite effective.

4.1.3 Linear reasoning in normed spaces

Formalising complex analysis needs a lot of tedious lemmas about distances. Over a fixed dimension, we can take components and solve with real quantifier elimination. However, a linear statement about norms is nonlinear after this reduction. So generalise to normed spaces. Of course, not all such statements will remain true, but many will, and in the author's experience, do.

4.2 Towards Constructive Homological Algebra — TC

Work is based on formalising Kenzo, a system in Homological Algebra. It deals with infinite sequences: $\ldots d_n : A_{n+1} \to A_n \ldots$. $d_{n+1} \cdot d_n = 1$, and we are interested in $H_n = \ker(d_n)/\Im(d_{n+1})$. Easy if A_n are finitely generated, but this is the aim of most computations. So we have an infinite sequence of infinite objects. There is no independent check on Kenzo, so we would like a proof of correctness. But Kenzo is 16Kloc LISP.

Aransay has a proof of the Basic Puerturbation Lemma in Isabelle, and there is also work in ACL2. Rather than prove Kenzo correct, we wish to represent Kenzo *in* type theory. Gregoire and Leroy have improved proving in type theory, and Thery has proved the associative law for elliptic curves in type theory. If $K \subseteq G$ and $h : G \to G$, we also want to consider $h : K \to G$. This is not directly possible in higher-order type theory. So we use a representation of category theory in higher-order logic. At this level, the reasoning is almost equational. However, we have problems with a 'universe', and therefore have to go to pre-categories.

4.3 What might "Understand a Function" mean — JHD

`http://staff.bath.ac.uk/masjhd/Slides/Calculemus2007.pdf`.

DS asked about Riemann surfaces. JHD said that these are very useful for humans, but that he did not know how to represent their generality in algebra.

RR asked about the relevance of his algorithm for integrating rational functions over the rational numbers. JHD said that this, very useful, algorithm produced an expression for the integral that had no *unnecessary* branch cuts. This in itself was a great step forward. RR insisted on *rational numbers*, and said that Maple and Axiom applied it outside this area. JHD thought that this was an instance of the problem of embedding algebraic numbers (ex Trager–Rothstein) in **C**.

4.4 Biform theories in Chiron — WMF

There are two main approaches to mechanized mathematics: proof and algebra. This can also be thought of as 'axiomatic' versus 'algorithmic'. In an algorithmic theory, the background assumptions are "in the background". There is no clear division between what is assumed and what is derived. Mathscheme is an integrated approach, not by linking two existing systems, but by producing an inetgrated framework. From this point of view, it is similar to Theorema.

We use biform theories: simultaneously both axiomatic and algorithmic. In fact it is a network of small inter-connected theories. We need a method of trustable communication. Let $L = (\mathcal{E}, \mathcal{F})$ be a anguage. An n-ary transformer is $(\pi, \hat{\pi})$ wiere π is a symbol and $\hat{\pi}$ is an algorithm implementing a (possibly partial) function $f : \mathcal{E}^n \to \mathcal{E}$. Associated with transformer we may have "meaning formulae".

A biform theory is $T = (L, \Omega)$. T is an axiomatic theory *and* al algorithmic theory. Chiron (the one good centaur) incorporates von Neumann–Bernays–Gödel set theory, elements of type theory, the 'traditional' approach to undefinedness and quotation/evaluation. Sets, classes and superclasses, T and F (which are not even superclasses), bottom (which is none of the above).

Every expression e has a semantic value e, and a syntactic value `(quote` e`)`.

4.5 Rational Reconstruction of a System for Experimental Mathematics — VS

We had a procedure for generating novel classification theorems in simple, finite algebraic structures. Currently integrate 15 different software software systems. The integration is very *ad hoc*, but it seems to work, and produce new results.

Now that we've done the case study, can we understand what we have done? So we wish to analyse the system by clearly distinguishing syntactic and semantic features.

Generate a quasi-group Q_1 of size 3. Ask a SAT solver 'are all quasigroups isomorphic to Q_1?' Answer is no, so generate a different Q_2. **Generate** a property P_1 that distinguishes Q_1 from Q_2. Prove that it's an isomorphism invariant. Prove that all 'quasi-groups with P_1' are isomorphic to Q_1. Then consider 'quasi-groups with $\not P_1$', and so on. So we are

1. specifying syntactically semantics of mathematics
2. which manipulate the syntax of mathematical theories
3. which are inhabited by mathematical objects.

Hence biform theories (but not Chiron itself).

Concentrated on one sub-process (in bold above), the machine-learning of isotopy-invariants for loops, which are Latin squares with a unit. Having generated a candidate, we have a bunch of theorem-provers which *might* prove that it is an isotopy-invariant. In biform-speak, we have seven transformers on three levels. This is formalised in second-order type theory. $\pi_1(A, C) = \mathtt{Prove}(A, C)$. $\hat{\pi}_1$ is a first-order automatic theorem prover. The corresponding meaning formula is $Proves(\mathtt{Prove}(A, C), A \vdash C)$.

So we can exhibit a clear separation between the generic and the domain specific. There is also scope for various optimisation of the process now that it is viewed as a series of generators and filters.

4.6 Automatic Synthesis of Decision Procedures: A Case Study of Ground and Linear Arithmetic — PJ

Decision procedures may be simple in principle, but in practice are large and error-prone. Many setps are "routine rewriting". Bundy's programme aims at identifying the common families of steps, such as 'remove', 'stratify', and many procedures also require quantifier elimination. An example of stratification would be reduction to CNF, which involves a certain orientation of rewrite rules. Although human assistance may still be required, the automatically-generated part should be sound. In linear arithmetic, there is 'cross-multiply and add' which requires a special-purpose generator. 'Thin' removes multiple unary operators (e.g. negation)

Why 'method generators' rather than 'methods'? So that we can combine them with other method generators to gain the strength of the system. Conditional rewrite rules are more complex, but still usable. We aim to produce a sequence of BNFs, each transformable to the next, which terminates in the trivial BNF: if so, we have a decision method. Searching over method generators is muuch more efficient than searching over individual rules.

Ground arithmetic has 59 rewrite rules, and we have 22 methods in a decision procedure. This took 3 seconds. For linear arithmetic we arrived, after 5 seconds, at a Fournier-Motzkin procedure in 51 methods. Method generators currently need to be told what sort of BNF to generate, and we would like to weaken this. However, we have already shown that automated (or semi-automated) generation of decision procedures is possible. These synthesised procedures apply rewrite rules in easily-understood stages.

TJ queried that the input already had the rewrite rules, and therefore how was commutativity handled? How does the system recognise that commutativity cannot be used indiscriminately. Answer: the meta-generator.

Part IV

MKM 2007

Chapter 5

Thursday 28 June

5.1 Let's compute a proof: aspects of proving with computer algebra — PP

This was a joint Calculemus/MKM invited talk.

He quoted DLMF and Sloane's Handbook as major marks in Mathematical Knowledge Management. He mentioned Zeilberger's 'holonomic systems approach to special function identities' approach. Can use computer algebra to produce proofs of previously unknown, but needed, identities. It is necessary to test for convergence, but this has not yet been fully automated. He stressed the importance of being able to manipulate inequalities in this context. For example (for real t, $m > 0$)

$$\sin(t)^{m+2} \leq \sin(t)^{m+1} \leq \sin(t)^{m} \tag{5.1}$$

on which Mathematica grinds for a while and then produces a very lengthy description of the truth conditions. This includes the $0 \leq t \leq \pi$ case. For some cases, the Zeilberger&... method can produce a formula which is a *certificate* of the theorem, which can then be validated independently of its derivation. This technology would lead to an automatic proof of Wallis's product formula for π, except that Maple/Mathematica are very por at limits of products of factorials.

Consider a lengthy summand which JHD could not get down, for which Mathematica returns 0, even though in fact only alternate terms are zero, due to an "initial conditions" problem.

He then looked at certificates of the Rogers–Ramajunan identities. The products can be converted into sums by Jacobi's triple product identity. The identities are first generalised, and then proved as sums which involve a recurrence. The recurrences are the same, so it is sufficient to identify the initial terms. In one case, we only have a recurrence of order five, whereas we might have expected order two.

Some of Slater's list of 130 RR-type identities are still a challenge for computer-aided proofs.

$$\int_0^\infty \frac{1}{(x^4+6x_1^2)^k} = \text{closed forms}$$

but Mathematica can take minutes to do even $k = 5$. Moll has an ingenious summation representation. There are routes using Ramajunan's master theorem, as corrected by Hardy. But the side-conditions are very tedious to check manually.

5.2 The Utility of OpenMath — JHD

`http://staff.bath.ac.uk/masjhd/Slides/MKM2007.pdf`. CSC distinguished three levels: notation (or presentation), content and logic. OpenMath, he thought, does well at distinguishing content from notation. He then asked whether `DefMP` wasn't mixing the last two — how can I interpret your `DefMP` if I don't know your logic. JHD admitted that thismight be a problem for type 4 symbols. Type 3 symbols have a purely extensional definition, so the logic used should be irrelevant.

5.3 Software Specification using Tabular Expressions and OMDoc — DP

Software specifications tend to involve a lot of mathematical notation, albeit generally of the "discrete mathematics" variety. If they are so powerful, why aren't they more used? Reading/writing them is hard, and keeping them in step with the code can be hard. Current tools don't add enough value to justify the effort required.

Can we do better? Differentiating content from presentation would be useful. We don't want to develop a new method as such. Tabular expressions (Parness it et al.) are designed to improve readibility and writability. A tabular expression for us is a list of grids, an evaluation term (how the grid is evaluated), static restrictions on the grid independent of expressions (shape), dynamic restrictions (disjoint and complete).

These mathematics come up in the sort of documents (specifications, module requirement etc.) used in software engineering. Since they are documents, can OMDoc help? We need theories (specifications), symbols, types, definitions (interested in `DefMP`) etc. Build on top of `www.eclipse.org`.

For one sort of verification we decided to use PVS, and used `omdoc2pvs.xsl` (MK). However, we are forced to use PVS type symbols (dependent types). This has found incompleteness cases. However, they are not currently imported PVS-proofs back into the system.

The conclusion is that this looks useful, and the tools are prvoding good leverage. Is this optimal, and will OMDoc 2.0, or OpenMath 3, change things.

PM-R said that these tables were ideal for scientific programming, and asked whether these concepts had been exported to other communities — not yet.

5.4 First Steps on Using OpenMath to add proving capabilities to standard dynamic geometry systems — MA

Tool is LAD — Locus, Assertion and Discovery. The dynamic geometry methods/systems (known as DGS) used are Cabri, the Geometer's Sketchpad and Cinderella. The DGS file and the problem spec. are converted by an applet into a DGS-neutral (OpenMath) file which goes to LAD, which uses Mathematica/CoCoA. LAD is a remote system — no computer algebra needed by the user. Cabri is numeric, Cinderella is probabilistic.

There are 6 CDs and 42 OMSs devoted to plane geometry, devloped by the Cinderella group at RIACA. Roozemond used these in 2004 ot add proving capabilities to Cinderella. No definitions for 'locus' or 'discovery'. Added a new OMS 'locus' taking the variables that trace the locus. (JHD: ?this should be an OMBIND. JHD recalls having this debate with the Cinderella group at Dagstuhl.) Also, a segment in `plangeo2` is not named, so not allowing cross-references, and so added this[1] to `<OMS name="segments" cd="plangeo2"/>`. Intergeo is a 3-year project starting in October 2007 to produce a geometric interlingua, based on OpenMath: it includes the Cabri and Cinderella teams.

The system *does* produce some non-degeneracy conditions.

OC invited the authors to join the OpenMath Society, since they clearly met the criterion. PI asked if there was any contact with the STEP group? The author did not, but thought there was within Intergeo.

5.5 Online Encyclopedia of Integer Sequences — NJAS

This was a joint Calculemus/MKM invited talk.

NJAS first demonstrated the OEIS and its webcam. There were originally (1964–1968) three punched cards per sequence, which still has vestiges in the current system.

You can't trust plots: the N'th term of the sequence of numbers which are precisely the product of three primes, is asymptotic to $\frac{2\log N}{(\log\log N)^2}$, which looks like a straight line but isn't. He also played the sequence of the number of groups of order n.

1973 was the Handbook (2500 sequences). At one point he had a cubic metre of material. The Encyclopedia (1995) had 5500 sequences. There are now 2^{17} sequences in the database. History of OEIS recapitulates CS: hardware, editors (`ed` beats punched cards, now emacs), systems, CGI scripts, working from postcards. The main database is still a flat file — 120Mb. The shortest sequence is A76337 (Riesel Numbers), of which only one is known. One major

[1] The CD is `experimental`, so the presenter says this is allowed.

problem is distinguishing conjectures from proofs. The curling numbers[2] are unbounded, but 4 appears at 220, 5 at about $10^{10^{23}}, \ldots$.

Some people do 'reverse engineering' on sequences, which is useful. His 'superseeker' program will discover if a sequence is, e.g., differences of another sequence, but (for combinatorial reasons), not combinations of sequences. Some questions are hard: "$\lfloor H_n + e^{H_n} \log H_n \rfloor - \sigma_n$ always non-negative" is equivalent to the Riemann Hypothesis.

[2] Start 1,1,2,1, then write as XY^k with k maximal (X potentially empty, Y not), and the next number is k. So $1, 1, 2, 1, 1, 2, 2, 2, 3, \ldots$.

Part V

Calculemus 2007 continued

5.6 Property Inference for Maple: An Application of Abstract Interpretation — JC

What properties can we find in 1Mloc? Maple has *all* of the following:

1. Imperative and Functional Programming;
2. Dynamic Typing (can encode a 'halts' type!);
3. Polymorphism: ad hoc, parametric and intensional;
4. Reflection and Reification;
5. Dependent "types";
6. First class "types".

Clearly insoluble, so what can we do that is finite and sound? There is a concept of a 'surface type', i.e. root node. Sequence are variable length, but lengths can generally be bounded. How many times is a variable read (0='useless', 1='can be inlined') or written (0='symbol')? A variable that cannot be typed *is* an error. `Digits` is a hidden parameter to many functions.

The basic philosophy is abstract interpretation. The first pass collects constraint information, in a rich constraint lattice, which includes recurrences on lattices. In practice most recurrences induced by loops and recursion are simple. In the second pass, we try to solve the constraints, and this solver is currently *ad hoc*.

Analysed 1276 procedures, 862 of which produced 'useful' information. Of 1330 locals, 721 had a single type. Uses Maple itself to sole recurrences etc. Need to feed the results of these analyses through in to other tools like mint.

5.7 Towards practical reflection for formal mathematics — MG1

Mathematical Theory Explanation consists of invention of concepts, invention of propositions about concepts, and proof of propositions, but there is also investigation of algorithms, new techniques etc. Theorema should allow one to do all of this in *one* language. But most systems don't allow one to add new reasoners, or reason about reasoners.

Our system is MiniTma. We interpret equational Horn clauses as rewrite rules. But we are hoping to use a pure SML-like fragment compiled to Java. Underlining represents quoted symbols. We sugar $\underline{\texttt{f[a,b]}}$ for $\underline{\mathrm{f}}[\underline{\mathrm{a}}, \underline{\mathrm{b}}]$. We use explicit quantification: $\underline{\texttt{forall}}[\underline{\mathrm{x}}, \underline{\mathrm{p(x)}}]$.

Need term induction to do logical reasoning, inducting on both depth and length. What about reasoning about reasoners? The most obvious result is soundness of a reasoner. However, we might also want, e.g., normalisation results or completeness results.

There are some foundational issues about reflection: most past/present (WMF) studies concentrate on a single fixed logic, which is not the case with Theorema. The Boyer–Moore ACL2 is the closest in terms of soundness proofs by reflection.

JC asked why not compile SML to SML? Partly efficiency reasons, but mostly because Mathematica has a Java interface.

5.8 The efficiency of geometric theorem proving by means of Gröbner Bases — SM

Motivation: Chou 1988 and the collection of 512 theorems proved by Wu's method. 477 of them were also proved by Gröbner bases: what of the rest? We use Maple 11 (Gröbner bases, `grevlex`) and the Epsilon library of D. Wang (characteristic sets). With this we have proved 26 (using some new techniques). 9 seem to have intrinsic space complexity problems. We also have what we believe to be the first Gröbner-base proof of Thébault–Taylor.

We view the hypotheses as $f_i \in \mathbf{Q}(u_1, \ldots, u_m)[x_i, \ldots, x_n]$. We ask if conclusion(s) $\in \sqrt{(f_1, \ldots, f_k)}$. We have to worry about non-degeneracy conditions in the u_i. 9 of the 26 required manual processing of these conditions. The unsolved ones are 6,7,10,11,12,13,14 (Pascal's theorem and variants), 19 (Briançon's Theorem) and 80 (Pratt–Wu).

$\mathbf{Q}[u_i][x_i]$ is much slower than $\mathbf{Q}(u_i)[x_i]$, but `fgb` in Maple is not available for the latter.

A questioner asked "why Gröbner bases if characteristic sets work"? BB replied that Gröbner bases were a decision procedure.

5.9 Future Directions for Calculemus and MKM

A panel session chaired by WMF. Panelists: BB, TC, FK, MK, AT. WMF started by saying that there were many topics for discussion, not least the future relationship between MKM and Calculemus.

MK He would confine himself to the research questions. Much of what we have seen is "metamathematics in the large". MathML is concerned with "mathematics in the small": in short all between dollar signs. But there is more to mathematics than individual formulae. What about internet-scale mathematics?

TC "Mathematics, Algorithms and Proofs" has had funding for five years. Its goals are similar to those of Calculemus. There have been meeting in Dagstuhl, Luminy and Leiden. Next year there will be a summer school in Trieste. He spoke about recent work about the connection between computer algebra and proof theory.

FK Frege worked alone, re-defining mathematics on the basis of logic. We know what happened to his work. The first half of the last century was full of new theories, category, set etc., first-order, higher-order etc. Then the computer came. She mentioned de Bruijn, AT and BB. How can we build things *together* using this technology? There is the 'Alexandria Library' project. We are doing more than computerising mathematics: we are digitising a whole area. Our systems must be tolerant of each other.

BB said how happy he was to see MKM and Calculemus here at RISC. This exemplified his vision in the initial editorial of the *Journal of Symbolic Computation*. He had hoped for more examples like Collins' Cylindrical Algebraic Decomposition. The Caculemus project had much the same aims. The inspiration of (MK)M was a paper with Hazewinkel about M(KM). There is necessarily a great deal of algebra and theorem-proving, but there is also much to be done on the organisational level of knowledge. He feels that the two are not the same, but the intersection is non-trivial, and co-location has clearly worked. However, should we also look at bringing them closer to ISSAC and IJCAI?

AT The Mizar group has been running for over thirty years. A datasbe of mathematics might take over 500 years to build. But in proving properties of programs, we need more than programming, we need mathematics, but not all of mathematics. Is this fragment easier?

A speaker mentioned an "empirically successful empirical reasoning over large theories" workshop, which has apparently been successful, though there is an anti-XML bias. This workshop has a most interesting competition. "We should keep our friends close, but our enemies closer". RM spoke from the point of view of an MK organiser. He felt that co-location had clearly worked. There were many out-of-bounds papers, which were pure "knowledge management", but we don't seem to be leveraging from this. WMF commented that "knowledge management" was a very large field. VS is chairing AISC next year, and he commented that it also lived in the 'Venn Diagram' that BB had put up. BB added that AISC had been in this very room in 2004, and its business meeting had asked 'what is AISC'. CSC said that the number of submissions had doubled, but not the number of people. MMK said that there had been some growth, but doubling would be surprising. He felt that co-location did lead to more people present.

MK said that one problem was that we are not really being noticed by the mathematicians. The Joint Mathematics meetings in North America have been successful, but there isn't the equivalent in Europe. There are also issues of our communication with the representers in other sciences. In his opinion, we have insights that should be shared with other communities. SMW, drawing on his experience of various communities, thought that the balance at this meeting was an improvement on some others.

BB said that, as a working mathematician, he was waiting for MK tools. DP said that the amount of mathematics being published was overwhelming, so most mathematicians need some KM tools. MMK said that it was unrealistic to hope for “the answer”, and echoed FK’s call for inter-system tolerance. He also recalled the success of the Calculemus Autumn School. MK added that the less formal sessions had had more industrial involvement, and we should do more of this.

CSC was not convinced that other scientists had the same problems, but MK disagreed. CSC argued that the clear layering distinction of mathematics was not present elsewhere. MK said that geographers had the same problems as us of distinguishing semantics from presentation. CR asked “where does mathematics stop”? Apparently the workshops mentioned above have “non-mathematicians”. FK cited Leibniz. JC said that theorem-provers are too slow and too hard to use, but computer algebra systems were being abused to use analysis for which they were not directly suited. Surely the two communities had to talk to each other?

SMW asked if we had locations for MKM and Calculemus 2008. Apparently we do. He argued for co-location next year, as a prelude to a permanent decision. MK said that VS had offered[3] to co-host other conferences with AISC in Birmingham. Calculemus had apparently decided to go there in 2008.

[3] VS confirmed this directly to JHD.

Chapter 6

Friday 29 June

6.1 A Collection of Elementary Problems in Geometry — TCH

This was a joint Calculemus/MKM invited talk.

Conjecture 1 (Kepler) *No packing on congruent balls in three dimensions can have density greater than* $\pi/\sqrt{18}$.

Proved by TCH in 1998. There seem to be many critical points (local minima) in this problem. The aim of the Flyspeck project is to give a Formal Proof of Conjecture of Kepler.

> The referees [aparently twelve of them] put a level of energy into this that it, in my experience, unprecedented. They ran a seminar on it for weeks. [...] They have not certified the proof, and will not be able to do so because they have run out of energy.[1]

This experience led to a new statement on computer-assisted proofs by Annals of Mathematics. The following contributions have been made to the project.

1. JH has released HOL 2.0
2. Bauer–Nipkow on the clasification of tame planar graphs.
3. Zumkeller (in Coq[2]), McLaughlin on nonlinear inequalities.
4. Obua (Isabelle) on formal verification of linear programming — there are over 1000 in the proof, which in 1998 were not fully automated.
5. The proof of the Jordan Curve Theorem.

[1] PI subsequently pointed out to JHD that this was actually the "disclaimer" printed by Annals of Mathematics with TCH's paper.

[2] This is currently still too slow for the real problem.

The optimisation problem can be factored into a nonlinear low-dimensional part and a high-dimensional linear part.

The 300 pages of text have expanded to 450, starting with 20 pages of trigonometry, such as the following.

Lemma 1 (1.36: Law of cosiness) *Let .... Then*

$$\cos\gamma = \frac{\cos c - \cos a \cos b}{\sin a \sin b}.$$

Chapter 2 (25 pages) is about the volumes of a variety (about twelve?)of three-dimensional figures. Chapter 3 (30 pages) is planar graph theory. TCH is surprised that the four-colour theorem, which is more about the plane than the Kepler Conjecture is, could be proved without the topology provided by the Jordan Curve Theorem. He has now imported some of the combinatorial underpinning of the four-colour theorem into his proof.

The main body will be about 100 pages of 'legal mathematics' (i.e. formalisms). There will then be 35 pages on tame graphs, 40 pages of linear programming, 10 pages of monotoniicty/derivatives, 10 pages of linear assembly, 130 pages of problems in geometry, and then the nonlinear inequalities.

In the geometry, every problem (of which there are about 150)[3] is Tarski (hence *in principle* decidable), involves a small number of variables (but a tetrahedron is 12 variables, so 'small' is relative) and deals with a configuration in $\mathbf{R}^3$. There are about 20 concepts defined (collinearity etc.). Some of the more interesting ones are about the three-dimensional equivalent of 'acute'/'obtuse', which can be characterised in terms of the polynomial χ. There is a collection of Cayley–Menger results.

There are various theorems of the form "a quadrilateral with long sides cannot have short diagonals". Many reduce to quantifier elimination problems, e.g. in fourteen variables (can be reduced to six).

In conclusion, the (current) division of the text into chapters has made it easier to see where to begin in the formalisation of the text, whereas originally it was only the programming that interested the verifiers.

DS: is there anything that Archimedes did not know? TCH: "yes, but not much" (JHD's interpretation[4]).

MK thought that there was a real MK problem here — how did one manage the knowledge required for such a PM-R raised the "13 sphere" problem. TCh said that there are approaches to the Kepler Conjecture that use this *en route*. "13 spheres" can be proved via semi-definite programming. AS asked whether the referees would find the re-organised version more accessible. TCH felt this was so, and that formalisation did not imply complication. There are questions of how can one trust so much formal mathematics: apparently Harrison/Plotkin have an implementation of Tarski in HOL.

[3] TCH referred JHD to the "2007 updates" on his web page. He later agreed that classical QE procedures did not handle segements well.

[4] Later corrected by DS to be "nothing except Euler's formulation of spherical triangles".

6.2 Rule-Based Simplification in Vector-Product Spaces — DJ

> An algebra or analytical method in which a single letter or other expression is used to specify a vector may be called a vector algebra. [Gibbs, c. 1890]

But most algebra systems do component analysis rather than true vector algebra. In components, can you tell $a \wedge (b \wedge c)$ from $(a \wedge b) \wedge c$? In vector algebra, we have both commutativity and anti-commutativity, associativity and anti-associativity, and so on.

Stoutemyer (1979) was the first (and only) to try this, in Macsyma. However, he could not get full simplification, because the system could not recognise that sub-terms were scalars, even though built out of vector constructs, and therefore the scalar simplifications were applicable.

So we really need an axiomatic treatment $\mathcal{T}$ of vector objects. The operators are $+, *, \bullet$ and $\wedge$, with $(abc) = a \wedge (b \wedge c)$. The additional axioms are:

1. $(a + b) \wedge c$;
2. $a \wedge b = -b \wedge a$;
3. $a \wedge (b \wedge c) = (a.c)b - (a.b)c$;
4. $(abc) = 0 \Rightarrow$ linear dependence.

From this system,we deduce about 10 auxiliary rules. Why is this axiom system three-dimensional? This formalism forces 3 (or 1 or 0! — JHD: in characteristic 0): an alternative formalism also allows 7.

There is an implementation in Aldor.

Definition 1 (Normal Forms) *A normal form is a sum of terms. A term is coefficient times scalar part times vector part. In the scalar part, we have scalars, scalar products or triple products.*

This representation solves Stoutemyer's problem.

TC: why not Clifford algebras? DJ — we probably should. JH pointed out that there was a Clifford algebra system in MatLab. RR added that there was one in Axiom.

6.3 Quantifier Elimination for Approximate Factorization of Linear Partial Differential Operators — [SMC presented by WW]

Such an operator is in $F[\partial_x, \partial_y]$. Second order operators $a_{2,0}\partial_x^2 + a_{1,1}\partial_x\partial_y + a_{0,2}\partial_y^2 + \cdots$ can be classified: hyperbolic iff $a_{1,1}^2 - 4a_{2,0}a_{0,2} > 0$, elliptic if < 0 etc.

Factorization of LPDOs is not well-understood, and multiplication is in general non-commutative. In particular factorisation is not unique. There is a method of searching for first-order factors from the left. If we write down a symbolic factorisation, we get a necessary and sufficient factorisation condition. Approximate factorization would be "to within ϵ in a bounded domain". We restrict the rest of our attention to problems linear in x and y.

Used QEPCAD-B. Needed to reduce the number of variables, e.g. by fixing ϵ and the bounds. 4Mb did not work, and increasing to 80Mb did not help. On a cut-down version, it did complete after 191 seconds — see proceedings. QEPCAD can produce simpler results by being given three different cases $a > 0$, $a < 0$ and $a = 0, b \neq 0$. He believes that this result is the simplest one for this problem.

JHD was asked to answer the questions! He commented that he was not surprised by the fact that moving from 4Mb to 80 did not help.

6.4 Calculemus Business Meeting — VS

Yesterday the trustees suggested that Programme Chairs for next year would be Rubio and Wiedijk — the latter had yet to be approached. The trustees would like to colocate with AISC 2008 in Birmingham, with VS as local organiser. The original dates were last week of July, after ISSAC in Hagenberg. However, some people would prefer *before* ISSAC.

The Trustees proposed that the charter be amended to arrange for one of the trustees to be designated as General Secretary — this was carried *unum contra*.

There are vacancies for four trustees (the term is three years). The Committee nominated JHD, who signified his willingness, and TC, who signified his willingness after some arm-twisting. Other candidates were discussed. The nominations will close in two weeks, and an e-mail will be sent out.

6.5 Mathematics and Scientific Notation — PM-R

This was a joint Calculemus/MKM invited talk.

> Power corrupts, and Powerpoint corrupts absolutely. [Tufte]

He had a collection of questions/points.

- Can a robot read a scientific thesis as currently produced? Not as well as external examiners, but not too bad.
- Can we give each o.d.e. a unique identifier? In theory no, in practice possibly.
- Authoring tools are the force for a destructive revolution.

CML and MathML/OpenMath are united in the importance of dictionaries. Crystallography had a dictionary, which inspired PM-R. Every new release of the browsers breaks things, and there are legacy problems, especially bitmaps.

On the other hand, Mathematics is poor, while Chemistry is rich, but doesn't spend on this. There is almost no open-access in Chemistry publication.

His team now has an XML-based chemical semantic framework for processing data, be it by Fortran IV programs or a PERL lash-up. He pointed to `DBpedia.org`, which has made Wikipedia into RDF (approx 10^9 triples). There is a query system called SNORQL (ouch!). Bioclipse is an exciting variant on Eclipse, which "knows about" molecules and spectra.

His students have written OSCAR, a program that can act as a 'robot referee' — "there are more hydrogen atoms in the structure than there are in the NMR" — and extract tables, spectra etc. PubChem is a tremendously valuable public[5] resource, which OSCAR can look at dynamically. It has InChI — machine-generated canonical identifiers for molecules.

He spoke about the Chemical Blogosphere. This contains a number of focused blogs, which are starting to use InChIs. The 'blue obelisk' dictionary of (chemical) algorithms contains MathML.

PI asked who was the keeper of CML — essentially PM-R. The Royal Society of Chemistry *does* believe in CML.

[5] American Chemical Society — "a socialist act by the Goverment which is destroying private enterprise". ACS refuses to quote InChI in its publications, but Nature now is.

Part VI

Programming Languages for Mechanized Mathematics

6.6 Mei — A Module System for Mechanized Mathematics Systems — JX

Module systems can be ML-like (structures, signatures, (higher-order) functors, type matching etc.) or Algebraic (specifications, union, renaming, first-order parameterized specifications, fitting morphisms etc.). We would like both, especially higher-order functors and fitting morphisms. This gives the following design goals.

- Should be independent of the language and logic of the underlying MMS.
- The theory is the basic uniit of mathematical knowledge.
- "little theory" approach: need to import a (translated version of) one theory into another.
- There are theory operations that build new ones from old ones. Extension, renaming and union are obvious ones.
- Support for generic modules (functors), which can be higher order.
- Functors are applicative — no context.
- A type and subtype mechanism.

He is largely working on the specification, but has a 'toy' implementation.

6.7 Algebraic Structures in Axiom and Isabelle: a Comparison — CB

Signatures are the interface specification mechanism for modules. By Axiom, he largely meant Aldor. Types are first-class objects in Axiom. Axiom is object-oriented. Categories and what Axiom calls signatures. Axioms in Axiom are essentially comments, but part of a contract that the implementor *should* honour.

Isabelle is based on the intuitionistic fragment of higher-order logic. HOL and ZF are two object logics. Types are not first-class citizens, and are too weak to express algebraic structures. We therefore have to use locales on top of Isabelle.

Specification for 'Groups acting on Sets' in both. On comparing both, we see that in Axiom, the parameters are arguments, whereas in locales, they are the signatures. The only morphism in Axiom is inheritance, whereas locales have several.

SMW pointed out that Group etc. in Axiom are constants, not (nullary) functions. RR pointed out that the action of a group on a set should not be a new object in Axiom. RR added that it would be useful to import 'associative', rather than having to re-specify it each time.

JC asked SMW what was stopping one having specifications in Axiom. One could have 'categories' like commutative. However, this would merely be declarative. JHD added that this would also block the way for a more far-reaching reform. SMW added that the Axiom method hard-coded the names of operators, so we have two towers, one each for additive and multiplicative groups. This also meant that one could not belong to a category in two different ways. Did Isabelle have this problem? CB replied that axiomatic type classes were too powerful. Hence locales provide essentially a 'views' ability. FW pointed out that he had once confused Coq to the point where his proof of associativity of multiplication was printed with '+'. An unsolved problem was sharing substructure in the presence of inheritance — JHD quoted 'I want the leading term operation from the `tdeg` view of this polynomial ring'

The RISCy boys

Entertained us this evening, with "Bookie Mountain" on drums and clarinet.

Chapter 7

Saturday 30 June

7.1 PML — a new proof assistant — CR1

Current proof assistants have liminations: module systems, weak equational reasoning etc. His view is that a proof assistant should start by being a programming language. He starts from a programming langauge (in his case ML-like), and will turn it into a deduction system, inspired by HOL. Start with a constraint consistency checking system. One problem is that the type system is too complex for the user. So we see the type system as a black box, which recovers types from programs.

How does it compare with ML? It has records, which can encode tuples, objects and modules. Functors are just functions Errors and exceptions are built in (errors cannot be caught). No type annotation is needed (except for multilpe inheritance, or for reasons of speed of checking). Polymorphism is very ML-like. Because of his ability to 'decorate' items, red-black trees can be represented as a derivative of trees. He needs/wants propositions and programs, but they *may* not be computable, e.g. with a universal quantifier. But, when `bool`=`prop`, it is easy to lose consistency. Hence he is moving to a version where proposition are sets.

We do not need abstract types, *if* we have specifications. HOL's ν and μ fixed-points are inconsistent over types. CR1 has a way of avoiding Russell's paradox. Because a program is finite, if its definition[1] order $\succ$ is non-cyclic, the program is well-founded. There is a `loop` error which can be raised by functions which *appear* not to be terminating. Can use external checkers to provde termination. We seem to need three sorts of proof: truth of a proposition; type refinement (proving that the type is smaller, e.g. technical constructors do not appear in the final output); proof of termination. But in fact one sort is enough: that an expression reduces to a value.

In the course of one example, he stated "I don't have polymorphic equality

[1] JHD asked if this forbade mutual recursion. No, 'definition' here doesn't mean textual definition, rather 'proven correctness'.

yet". He showed another example, in which he had to add a proof of termination to make a program 'acceptable', i.e. cannot raise the `loop` error. He needs a construct `let try` which allows one to catch exceptions in the first part of an evaluations.

There is a dependent type problem, since PML's type checker cannot handle the requirement for a correctness proof. This can (?sometimes) be solved by adding auxiliary types for "those x for which it succeeds"

The main ingredient of type checking in PML is Knuth–Bendix. In the current state of PML, Knuth–Bendix terminates[2]. He has some ideas on extending this.

The system is currently 11Kloc, he estimates <30 for the final system. He still needs: quantification via choice operators and exceptions; a resolution of the difference between Liebniz and computable equality; a termination check (in progress); macros and tactics; an investigation of the theoretical strength and comparison with Quine's NF and Jensen's NFU (which is equi-consistent with Z, not necessarily ZF). He believes (a formal proof needs to be done) that PML is equi-consistent with NFU.

7.2 Declarative Representation of Proof Terms — CSC

For example, a proof language may be much richer than the underlying logic. The logic may be the same as another system, but the proof will not be portable (different tactics, heuristics etc.). Of course, we want a genuine translation, not reducing all propositions to 'true'. We might also wish to move from a procedural proof language to a declarative one.

We know that both declarative and procedural proof languages can be compiled into λ-terms. We also know that (pseudo)-natural proofs can be converted from/to $\tilde{\lambda}\mu\tilde{\mu}$-terms, a larger prof-term space. CSC now claims that 'declarative' can be viewed as equivalent to 'pseudo-natural'. One direction follows from $\tilde{\lambda}\mu\tilde{\mu} \supset \lambda$. So what we need is λ-terms to 'declarative'. He presented an interesting tabular comparison.

He starts from the language used in MOWGLI. For reasons of space, he used a simpler language than ICC, but it has to have λ-abstraction over types as well as functions. Uses a 'small step' operational semantics, essentially a function from partial proofs to partial proofs. He has 'quite a simple' conversion from terms to statements.

[2] His orientation of rukes deducedfrom critical pairs is based on the fact that he wants expressions to reduce to values.

7.3 Procedural Representation of CIC Proof Terms — FG

[A historical note — the only talk to use an OHP.]

FG reminded us that CIC is the representation of Coq. This process has been applied to a Coq libary of over 600 proofs. A proof has 'contents' and 'structure'. From this point of view, 'declarative' is 'contents-oriented', while 'procedural' is 'structure-oriented'. Adapting a proof means changing the contents while maintaining the structure. Equally, analogies between proofs tend to be analogies of structures. Therefore these are more evident in procedural representations.

In CIC, the 'type level' is declarative, but the 'term level' is the procedural information. A tactic might expose either kind of information.

procedural `apply`, `cases`, `pose` v `as` x, `elim` v `using` t.

declarative `cut` w `as` x, `change` x `as` w.

Proof term optimisation improves the quality of scripts, not least by making them more readable. We can also convert proof-by-cases into proof-by-induction, which is not a CIC conversion. The conversion proof steps are detected by Coscoy's method, though this may not be the best method. Five lines becomes a full page, since the Coq inversion tactic inlines the inversion lemma.

7.4 What Happened to Languages for Symboic Mathematical Computation — SMW

Some people will consider this presentation to be trivially true, others trivially false. He has worked on Maple, which he considers to be a language with Lisp-semantics and Algol-68 syntax (others might disagree). Axiom/Aldor is a higher-order language with type categories and dependent types. A distinguishing feature is *ex-post-facto* extensions.

$$\textit{J. Symbolic Computation} \cap \textit{J. Lisp Symbolic Computation} = \emptyset. \qquad (7.1)$$

For SMW, computer algebra is arithmetic in *specific* structures such as rings, whereas symbolic computation, such as completing the square, is not in a given setting.

An embarrassment is $x^{n^2+n}-16$, which always factors (case n on even/odd), but no system can handle this[3]. SMW claims the following.

- Programming language support for computer algebra is good, and *can* be improved.

[3] DS pointed out in the questions that, if we replace n by $2n$ or by $2n+1$, then Mathematica *can* do the factorisation. Of course, DS had done the `case` himself.

- Programming language support for symbolic computation is poor, and *must* be improved

We need much better support for typed terms. Having defined `Integer`, we should be able to declare `a:Symbolic Integer;` and so on.

JC Are you saying that *the* thing we need is typed syntax?

SMW We need many things, but this is a basic block.

TC *typed terms* $\cup$ *NuPRL*.

SMW Where do we get the 200 man-years from? But you are right, no one group has solved this problem.

RR Focal etc. are trying to separate the computation from the logic.

SMW General agreement. Magma is, in a different sense, in the right direction, since algorithms, and their domain of applicability, are separated from concepts.

JC The aim of this meeting — "Mechanised Mathematics" — is to bridge this divide.

SMW It's hard/impossible to branch on a symbolic Boolean.

JH There is pressure in computer algebra for 'more' language features, but not, as far as he can tell, in theorem proving. One reason is that theorem provers (the human beings, tha is) are instinctively foundational, but there are also reflection issues.

SMW I would like 'write once, apply many times' semantics, which requires more power than Haskell's (very nice) monads.

7.5 SML with antiquotations embedded into Isabelle/Isar — MW

Isabelle is an infinite process, with theory development intertwined with programming. So what is the relationship between the logic and the programming language? Milner's answer — three languages: Implementation (Lisp)/Meta (ML)/Object Language (Logic). The logic can be LCF (DS) or HOL (Mike Gordon). In this approach we have the HOL family, Coq and Isabelle (Pure or HOL). The Implementation langugae is now the Meta language, either SML or OCaml. We now have simplified user languages, Coq vernacular, or Isar[4]. Coq has internalised computation — $\delta\iota$-reductions.

[4] DS later pointed out that Isar stood for "Intelligible Semi-Automated Reasoning", and asked whether antiquotations contributed to the intelligibility. MW thought that *selective* use did.

Isabeelle/Isar has SML as the implementation language, a primary layer (Isar) and a presentation layer (LaTeX). The primitive layers are the logic framework and application environments, and the embedded SML with antiquotations. In order to do this, we need run-time evaluations (see `EVAL` in Lisp). This has no return type, only side effects. This can be embedded to produce a value-returning form of `eval` (in a horrible, typeless but effective way). A quote is `<<...>>` and the antiquote is `@{name args}`. The following lets us introduce a theorem.

```
ML << val Ps: term List = Thm.prems_of @{thm mp} >>
```

This can be used to capture the ML-state, for debugging or program-understanding. We can use antiquotations to fix the type of certain variables and to allow run-time matching.

7.6 Computer Algebra and the three 'E's: Efficiency, Elegance and Expressiveness — JHD/JPff

`http://staff.bath.ac.uk/masjhd/Slides/PLMMS2007.pdf.`

SMW pointed out that he had a student trying to mechanise the 'special-case' compilation mechanism of Axiom. JH pointed out that the one area of theorem-proving that had benefited from 'machine-level' hacking was SAT.

Subsequently

JHD re-gave the talk to OC. She pointed out that Autexier's talk in MKM (which clashes) had something about the extension of operators to aggregates. User-defined notation would automatically be included. This system uses OM-Doc/OpenMath. She recommends downloading the system.

7.7 PLMMS Business

JC opened the discussion by asking "did we want another one" and, if so, in which format. JC asked why there were no FOC/Focal submissions. There are now other applications (than computer algebra) for Focal. If MKM/Calculemus/ /AISC really do co-locate, then it would be a good idea to go with them. SMW suggested that we should try this, and hope for more uptake. JH thought that we could get some invited speakers from, say, dependent types communities. JC said Simon Peyton-Jones had looked at why Haskell could not be used (effectively) to write computer algebra, which SMW confirmed. JC had also found problem with the module system in OCaml, which were welcomed by Xavier LeRoy.

There seemed to be general agreement to run with Calculemus etc. JC said that two co-chairs seemed reasonable. JH asked JC if he was willing to serve. JHD said that two co-chairs with desynchronised two-year terms were a plausible paradigm. JC said that, as long as some-one else did the local arrangements, he was willing. Various names were mentioned, and an ordered list produced for JC to work with. SMW said that it was slightly odd to ask some-one to chair a meeting that they had never been to.

After further discussion, SMW proposed JC and MW as co-chairs. JC recused himself as chair of the meeting, and JHD was chosen as chair of this business meeting. The motion was put to the vote, and carried *nem. con.*

Appendix A

Dramatis Personæ

AMC Arjeh Cohen (not present, but referred to).

AS Alan Sexton.

AT Andrzej Trybulec.

BB Bruno Buchberger.

BRM Bruce Miller.

BS Bernhard Stöger.

CB Clemens Ballarin.

CL Christoph Lange.

CR Chris Rowley.

CR1 Christophe Raffalli.

CSC Claudio Sacerdoti Coen.

DJ David Jeffries.

DP Dennis Peters.

DPC David Carlisle.

DS Dana Scott.

ES Elena Smirnova.

FG Ferruccio Guidi.

FK Fairouz Kamareddine.

FR Florian Rabe.

GG George Goguadze.

JC Jacques Carette.

J-FN Jean-François Nicaud.

JH John Harrison.

JHD James Davenport.

JPff John ffitch (not present, but referred to).

JX Jian Xu.

KM Klaus Miesenberger.

MA Miguel Abanades.

MCD Mike Dewar (not present, but referred to).

MG Marc Gaëtano (not present, but referred to).

MG1 Martin Giese.

MK Michael Kohlhase.

MMK Manfred Kerber[1].

MS Mika Seppälä.

MW Makarius Wenger.

NJAS Neil Sloane.

OC Olga Caprotti.

PI Patrick Ion.

PJ Predrag Janičić.

PL Paul Libbrecht.

PM-R Peter Murray-Rust.

PP Peter Paule.

RE Ramon Eixarch.

RM Robert Minor.

RR Renaud Rioboo.

RV Rikko Verrijzer.

[1] MMK admitted to JHD that he believed that computers were devices for telling you that the semi-colons were in the wrong place, until he met Reduce.

SD Sam Dooley.

SM Shuichi Moritsugu.

SMC Scott McCallum.

SMW Stephen Watt.

TC Thierry Coquand.

TCH Thomas Hales.

TJ Tudor Jebelean.

TK Temur Kutsia.

VS Volker Sorge.

WMF Bill Farmer.

WW Wolfgang Windsteiger.