

MathHub.info: Active Mathematics

Michael Kohlhase

—

with the KWARC group, in particular Florian Rabe

Computer Science, Jacobs University Bremen, Germany

<http://kwarc.info>

July 18. 2014, QED 20+, Vienna

1 Motivation: QED

My Recollection of (the end of) QED

- ▶ First QED Workshop in Argonne, May 1994
- ▶ The QED Manifesto published early in 1994 (worth a re-read)
- ▶ intensive discussion at CADE 1994 (inspired me greatly, influenced my thinking)

My Recollection of (the end of) QED

- ▶ First QED Workshop in Argonne, May 1994
- ▶ The QED Manifesto published early in 1994 (worth a re-read)
- ▶ intensive discussion at CADE 1994 (inspired me greatly, influenced my thinking)
- ▶ QED Mailing list, see <http://mizar.org/qed/>
(all my heroes were there, discussed all the important topics)
- ▶ Pertinent Topics
 - ▶ “Root Logic”, Calculi and Proof formats (most contentious issue)
 - ▶ Types and Partial Functions (that as well)
 - ▶ does Math need formalization or the other way around?
 - ▶ funding (never satisfactorily resolved)

My Recollection of (the end of) QED

- ▶ First QED Workshop in Argonne, May 1994
- ▶ The QED Manifesto published early in 1994 (worth a re-read)
- ▶ intensive discussion at CADE 1994 (inspired me greatly, influenced my thinking)
- ▶ QED Mailing list, see <http://mizar.org/qed/>
(all my heroes were there, discussed all the important topics)
- ▶ Pertinent Topics
 - ▶ “Root Logic”, Calculi and Proof formats (most contentious issue)
 - ▶ Types and Partial Functions (that as well)
 - ▶ does Math need formalization or the other way around?
 - ▶ funding (never satisfactorily resolved)
- ▶ The Root Logic Controversy
 - ▶ all: use my pet logic as the root logic please (all different for good reason)
 - ▶ Peter Andrews: Accept Diversity (re-read that e-mail)
 - ▶ Bob Boyer: use Recursive Peano Arithmetic (PRA) (executive order?)
 - ▶ Peter Andrews: If PRA, then we should Goedelize it first (end of consensus)

My Recollection of (the end of) QED

- ▶ First QED Workshop in Argonne, May 1994
- ▶ The QED Manifesto published early in 1994 (worth a re-read)
- ▶ intensive discussion at CADE 1994 (inspired me greatly, influenced my thinking)
- ▶ QED Mailing list, see <http://mizar.org/qed/>
(all my heroes were there, discussed all the important topics)
- ▶ Pertinent Topics
 - ▶ “Root Logic”, Calculi and Proof formats (most contentious issue)
 - ▶ Types and Partial Functions (that as well)
 - ▶ does Math need formalization or the other way around?
 - ▶ funding (never satisfactorily resolved)
- ▶ The Root Logic Controversy
 - ▶ all: use my pet logic as the root logic please (all different for good reason)
 - ▶ Peter Andrews: Accept Diversity (re-read that e-mail)
 - ▶ Bob Boyer: use Recursive Peano Arithmetic (PRA) (executive order?)
 - ▶ Peter Andrews: If PRA, then we should Goedelize it first (end of consensus)
- ▶ Second QED Workshop in Warsaw, July 1995

How can we do things better 20 years later?

- ▶ **Build on what we have:** Theorem Prover Libraries
e.g Mizar, HOL Light, Coq, Matita, IMPS, PVS, ...

How can we do things better 20 years later?

- ▶ **Build on what we have:** Theorem Prover Libraries
e.g Mizar, HOL Light, Coq, Matita, IMPS, PVS, ...
- ▶ **But:** this is **not QED yet**, since there is no
 - ▶ uniform resource for **all of math** (need logical pluralism & integration)

How can we do things better 20 years later?

- ▶ **Build on what we have:** Theorem Prover Libraries
e.g Mizar, HOL Light, Coq, Matita, IMPS, PVS, ...
- ▶ **But:** this is **not QED yet**, since there is no
 - ▶ uniform resource for **all of math** (need logical pluralism & integration)
 - ▶ **one-stop-shop** archive and user portal (need joint system)

How can we do things better 20 years later?

- ▶ **Build on what we have:** Theorem Prover Libraries
e.g Mizar, HOL Light, Coq, Matita, IMPS, PVS, ...
- ▶ **But:** this is **not QED yet**, since there is no
 - ▶ uniform resource for **all of math** (need logical pluralism & integration)
 - ▶ **one-stop-shop** archive and user portal (need joint system)
 - ▶ Math grows faster than the libraries (120 000 papers vs. 10 000? theorems)

How can we do things better 20 years later?

- ▶ **Build on what we have:** Theorem Prover Libraries
e.g Mizar, HOL Light, Coq, Matita, IMPS, PVS, ...
- ▶ **But:** this is **not QED yet**, since there is no
 - ▶ uniform resource for **all of math** (need logical pluralism & integration)
 - ▶ **one-stop-shop** archive and user portal (need joint system)
 - ▶ Math grows faster than the libraries (120 000 papers vs. 10 000? theorems)
- ▶ **Rest of this talk:** towards fixing these three problems.

Peter Andrews: Accept Diversity (email; 11 August 1994)

- ▶ [...] the need for choosing appropriate languages, proof systems, definitions, ways of organizing the library, etc. [...]
- ▶ [...] One may be led to despair when one thinks seriously about the prospects for achieving consensus on all of these matters. [...]
- ▶ [...] I envision the QED library as a vast database and a large number of associated utilities for accessing, displaying, translating, manipulating, and using the items in the database in many ways. [...]
- ▶ [...] Since it will be difficult or impossible to obtain universal agreement on the choice of a formal language, definitions and terminology, and proof systems, I propose that the QED library permit great variety in such matters. [...]

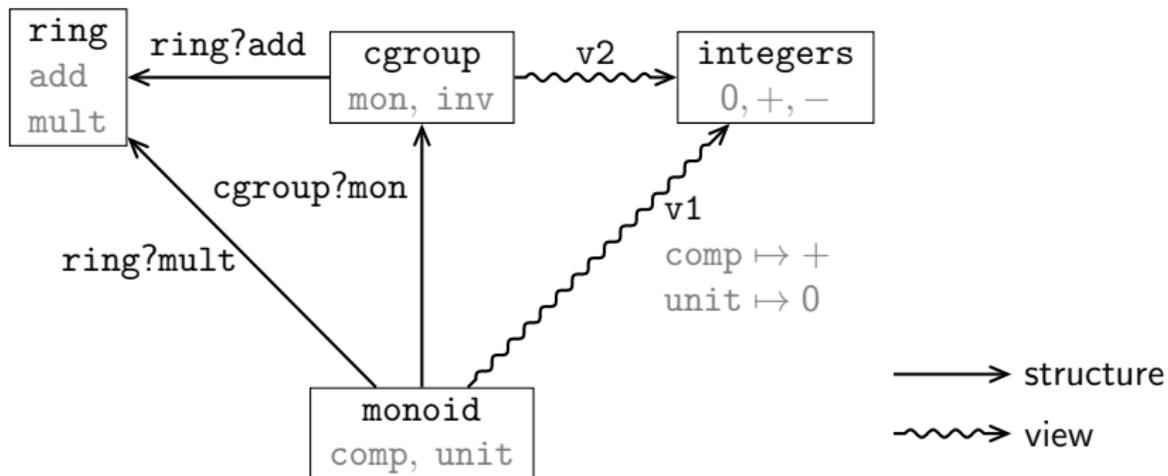
2 Logical Pluralism

MMT a Module System for Mathematical Content

- ▶ MMT: Universal representation language for formal mathematical/logical content
- ▶ **Implementation:** MMT API with generic
 - ▶ module system for math libraries, logics, foundations
 - ▶ parsing + type reconstruction + simplification
 - ▶ IDE (web server + JEdit)
 - ▶ change management
- ▶ Continuous development since 2007 (> 30000 lines of Scala code)
- ▶ Close relatives:
 - ▶ LF, Isabelle, Dedukti: but flexible choice of logical framework
 - ▶ Hets: but declarative logic definitions

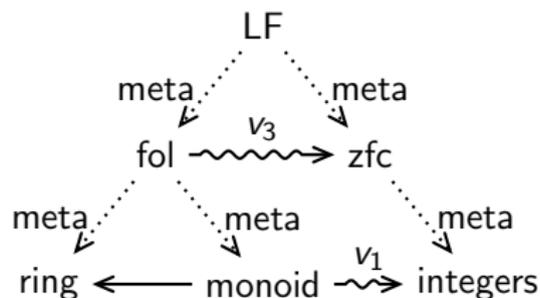
Example

$$\begin{array}{l} v2 \\ \left\{ \begin{array}{l} mon/comp \mapsto + \\ mon/unit \mapsto 0 \end{array} \right\} \text{ or } mon \mapsto v1 \\ inv \mapsto - \end{array}$$



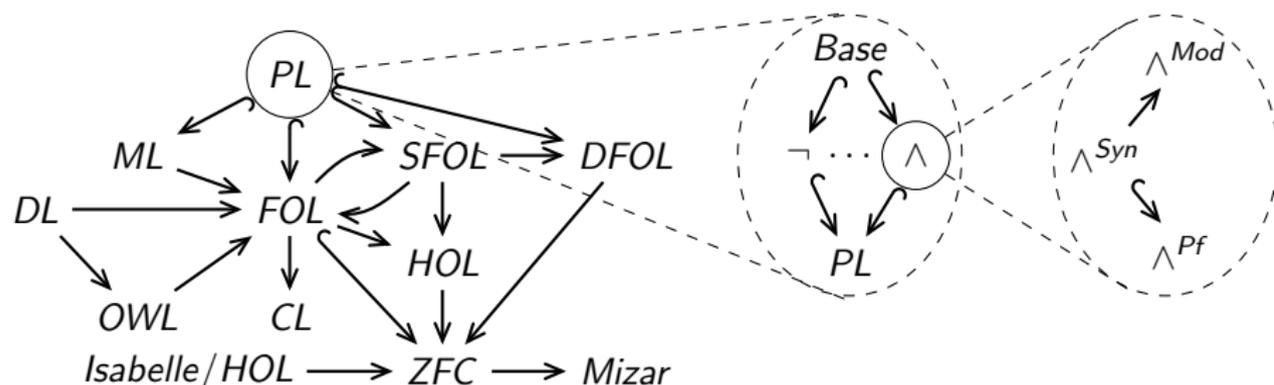
Representing Logics and Foundations as Theories

- ▶ Logics and foundations represented as theories
(subject to the same module system)
- ▶ Meta-relation between theories
(special case of inclusion)
- ▶ Semantics of logics represented as theory morphisms into the foundation, e.g., `folsem`
- ▶ Models represented as theory morphisms
 v_1 interprets monoid in integers using meta-morphism v_3



The LATIN Logic Atlas

- ▶ **Definition 2.1** The LATIN project (Logic Atlas and Integrator) develops a logic atlas, its home page is at <http://latin.omdoc.org>.
- ▶ **Idea:** Provide a standardized, well-documented set of theories for logical languages, logic morphisms as theory morphisms.



- ▶ **Technically:** Use MMT as a representation language **logics-as-theories**
- ▶ Integrate logic-based software systems via views.
- ▶ **State:** About 1000 modules (theories and morphisms) written in MMT/LF [RS09]

MMT API JEdit Integration (IDE)

The screenshot shows the JEdit IDE interface with the following components:

- Title Bar:** jEdit - C:\other\oaff\test\source\examples\pl.mmt
- Menu Bar:** File Edit Search Markers Folding View Utilities Macros Plugins Help
- Left Sidebar:** Project tree for 'pl.mmt' showing a hierarchy: theory PL > prop, ded, and, impl, equiv, type, definition, lambda, x, y, prop, and, impl, x, y, ded.
- Main Editor:** Contains MMT code:

```
1 namespace http://cds.omdoc.org/examplesGS
2 theory PL : http://cds.omdoc.org/urtheories?LF =
3   prop : typeRS
4   ded  : prop → typeUS # ded 1 prec 0RS
5   and  : prop → prop → propUS # 1 ∧ 2 prec 15RS
6   impl : prop → prop → propUS # 1 ⇒ 2 prec 10RS
7   equiv : prop → prop → propUS # 1 ⇔ 2 prec 10US
8   = [x,y] (x ⇒ y) ∧ dedRS
```
- Status Bar:** 1 error, 0 warnings
- Console Window:** C:\other\oaff\test\source\examples\pl.mmt (1 error, 0 warnings)
 - 8: invalid object: http://cds.omdoc.org/examples?PL?equiv?definition: ded
 - argument must have domain type
 - http://cds.omdoc.org/examples?PL; x:prop, y:prop |- ded : prop
 - http://cds.omdoc.org/examples?PL; x:prop, y:prop |- prop→type = prop
- Bottom Bar:** 8,30 (mmt,sidekick,UTF-8)S m r oWV 27.5 Mb 4 error(s)19:50

MMT API Browser Integration

The MMT Web Server
Graph View Administration Help

Style: html5 cds.omdoc.org / courses / 2013 / ACS1 / exercise_10.mmt ? Problem3

acs1_2013

- exercise_10.omdoc
 - Problem2
 - Problem3**
 - Problem4
- example
- latin
- lmfdb
- mathscheme
- mml
- openmath
- test
- tptp
- urtheories

theory Problem3 **meta** LF

include : <http://cds.omdoc.org/examples/FOLEQNatDed>

circ : term → term → term

e : term

R : ⊢ ∀ x ∘ e ∘ x

C : ⊢ ∀ x ∀ y x ∘ y ∘ y ∘ x

L : ⊢ ∀ x e ∘ x ∘ x

$$= \left[x \right] \frac{\frac{\frac{C\ e}{\vdash \forall y \circ y \circ x \circ e} \text{forallE } x}{\vdash e \circ x \circ x \circ e} \text{forallE} \quad \frac{R\ x}{\vdash x \circ e \circ x} \text{forallE}}{\vdash e \circ x \circ x} \text{forallE}$$

reconstructed types >
implicit arguments >
redundant brackets > show hide
infer type
simplify
fold

Enter an object over theory: http://cds.omdoc.org/courses/2013/ACS1/exercise_10.mmt

[x] x ∘ e

analyze simplify

[x] x ∘ e

{x:term} term



3 Flexiformality (to avoid losing the QED race)

Formalization in Mathematical Practice

- ▶ To formalize maths in a formal system \mathcal{S} , we need to choose a **foundation**, i.e. a foundational \mathcal{S} -theory, e.g. a set theory like ZFC.
- ▶ Formality is an **all-or-nothing property** (a single “obviously” can ruin it.)
- ▶ Almost all mathematical documents are informal in 4 ways:
 - ▶ **the foundation is unspecified** (they are essentially equivalent)
 - ▶ **the language is informal** (essentially opaque to MKM algos.)
 - ▶ even **formulae are informal** (presentation markup)
 - ▶ **context references are underspecified**
 - ▶ mathematical objects and concepts are often identified by name
 - ▶ statements (citations of definitions, theorems, and proofs) underspecified
 - ▶ theories and theory reuse not marked up at all
- ▶ The gold standard of mathematical communication is “**rigor**” (cf. [BC01])

Formalization in Mathematical Practice

- ▶ To formalize maths in a formal system S , we need to choose a **foundation**, i.e. a foundational S -theory, e.g. a set theory like ZFC.
- ▶ Formality is an **all-or-nothing property** (a single “obviously” can ruin it.)
- ▶ Almost all mathematical documents are informal in 4 ways:
- ▶ The gold standard of mathematical communication is “**rigor**” (cf. [BC01])
 - ▶ **Definition 3.3** We call a mathematical document **rigorous**, if it could be formalized in a formal system given enough resources.
 - ▶ This possibility is almost always unconsummated
 - ▶ **Why?**: There are four factors that disincentivize formalization for Maths
 - propaganda**: *Maths is done with pen and paper*
 - tedium**: de Bruijn factors ~ 4 for current systems (details in [Wie12])
 - inflexibility**: formalization requires commitment to formal system and foundation
 - proof verification useless**: peer reviewing works just fine for Math
 - ▶ **Definition 3.4** The **de Bruijn factor** is the quotient of the lengths of the formalization and the original text.

Formalization in Mathematical Practice

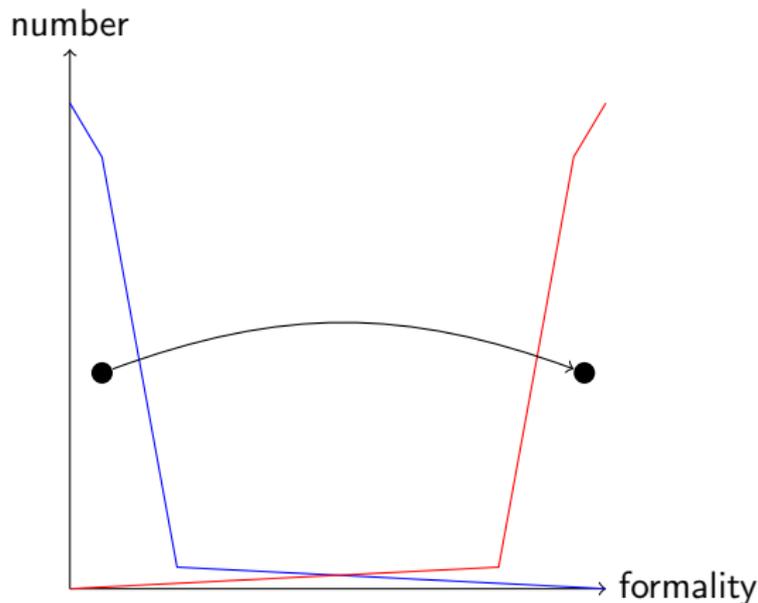
- ▶ To formalize maths in a formal system \mathcal{S} , we need to choose a **foundation**, i.e. a foundational \mathcal{S} -theory, e.g. a set theory like ZFC.
- ▶ Formality is an **all-or-nothing property** (a single “obviously” can ruin it.)
- ▶ Almost all mathematical documents are informal in 4 ways:
- ▶ The gold standard of mathematical communication is “rigor” (cf. [BC01])
 - ▶ **Definition 3.5** We call a mathematical document **rigorous**, if it could be formalized in a formal system given enough resources.
 - ▶ This possibility is almost always unconsummated
 - ▶ **Why?**: There are four factors that disincentivize formalization for Maths
 - propaganda**: *Maths is done with pen and paper*
 - tedium**: de Bruijn factors ~ 4 for current systems (details in [Wie12])
 - inflexibility**: formalization requires commitment to formal system and foundation
 - proof verification useless**: peer reviewing works just fine for Math
 - ▶ **Definition 3.6** The **de Bruijn factor** is the quotient of the lengths of the formalization and the original text.

In Effect: Hilbert’s program has been comforting but useless

- ▶ **Question**: What can we do to change this?

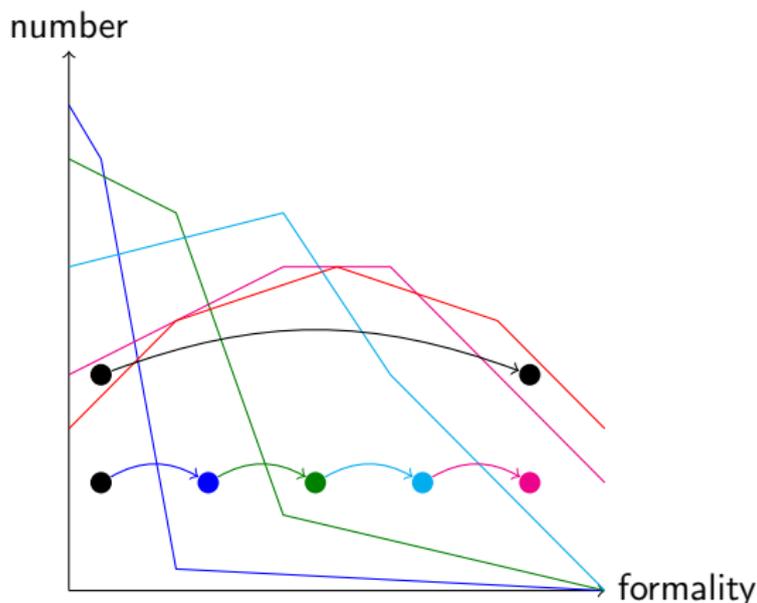
Migration by Stepwise Formalization

- ▶ Full Formalization is hard (we have to commit, make explicit)
- ▶ Let's look at documents and document collections.



Migration by Stepwise Formalization

- ▶ Full Formalization is hard (we have to commit, make explicit)
- ▶ Let's look at documents and document collections.
- ▶ Partial formalization allows us to
 - ▶ formalize stepwise, and
 - ▶ be flexible about the depth of formalization.



The Flexiformalist Program (Details in [Koh13])

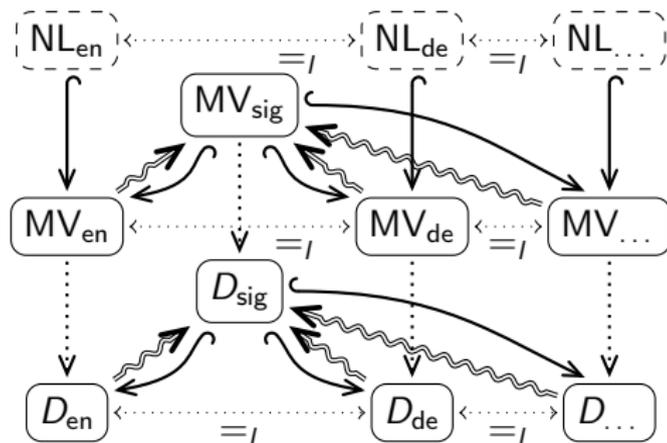
- ▶ The development of a **regime of partially formalizing**
 - ▶ **mathematical knowledge** into a modular ontology of mathematical theories (**content commons**), and
 - ▶ **mathematical documents** by semantic annotations and links into the content commons (**semantic documents**),
- ▶ The establishment of a **software infrastructure** with
 - ▶ a **distributed network of archives** that manage the content commons and collections of semantic documents,
 - ▶ **semantic web services** that perform tasks to support current and future mathematic practices
 - ▶ **active document players** that present semantic documents to readers and give access to respective
- ▶ the re-development of comprehensive part of mathematical knowledge and the mathematical documents that carries it into a **flexiformal digital library of mathematics**.

SMGloM: a Semantic Multilingual Glossary for Math

- ▶ Mathematical domains are characterized by their vocabularies.
- ▶ lexical information is at the center of parsing and understanding MathDocs.
- ▶ Lexical databases like WordNet [Wor] have kick-started language processing.
- ▶ **Definition 3.7** A **glossary** is a list of terms in a particular domain of knowledge with their definitions. (traditionally a list in the end of a book.)
- ▶ **Obviously:** Need a **glossary for mathematics** as a **semantic/linguistic resource!**
- ▶ **Related:** Similar things have been tried before
 - ▶ OpenMath Content Dictionaries (symbol-oriented, non-linguistic)
 - ▶ hundreds of elementary maths glossaries started (mostly discontinued)
- ▶ **Requirements:** The SMGloM must be
 - ▶ open and collaboratively curated (public resource)
 - ▶ flexiformal (to be machine-processable, but human-understandable)
 - ▶ multi-lingual (concepts are trans-lingual, but usage is language-dependent)
 - ▶ cover named entities and their syntax for words and formulae
- ▶ **State:** Planning, system development, writing seed content (stay tuned)

Implementing SMGloM in MMT: Mathematical Vernacular

- Using the MMT Meta-Levels in SMGloM: Introduce a new (meta-level) SMGloM module MV for mathematical vernacular (concepts like $=$, \forall , \exists , iff, hence, structures, piecewise definition)



Inheritance from natural language NL^* only at the conceptual level
(unrepresented in SMGloM)

4 Implementing all this (my personal QED dream)

MathHub Motivation: A shift in Focus

- ▶ **Problems and Focus Shifts:**
 - ▶ in **Theorem Proving/Formal Methods**: **libraries collapse under their own weight**
(towards corpus management and system interoperability)
 - ▶ in **Mathematical Knowledge Management (MKM)**: **individual services are boring**
(towards larger corpora)
 - ▶ in **Digital Mathematical Libraries (DML)**: **value added by mere digitization questionable**
(towards more semantics)
 - ▶ in **Mainstream Mathematics**: **how to deal with publication overload and the long historic tail**
(WHDML \rightsquigarrow to Sloan Funding?)
- ▶ **Common Theme**: large-scale, flexiformal, pluralist, digital libraries with machine support. (but nobody builds this!)
- ▶ **The Domain/Scope**: everything that is rigorous enough. (call this “Math”)
- ▶ **Idea**: learn from the open source community, offer a code repository with management support that acts as a hub for publication/development projects.
- ▶ **MathHub**: a collaborative development/hosting/publishing system of open-source, flexiformal math.

Requirements for MathHub: Representation

- ▶ MathHub must support
 - ▶ adequate representation of theorem prover libraries
 - ▶ formal and informal representations

Requirements for MathHub: Repository Management

- ▶ MathHub must support
 - ▶ hosted archiving with revision control
 - ▶ distributed development models
 - ▶ public and private development projects

Best-of-Breed Role Models: GitHub (Build software better, together)

(ca. 12 M Projects)

GitLab

(open source variant)

based on git for distributed revision control.

- ▶ **Additional Requirements for MathHub:**
 - ▶ syntax-highlighted source code is not enough.
 - ▶ active documents
 - ▶ content-based formula search
 - ▶ want to navigate domain relations and logic crosswalks

(embedded interactions)

Requirements for MathHub: Community Support

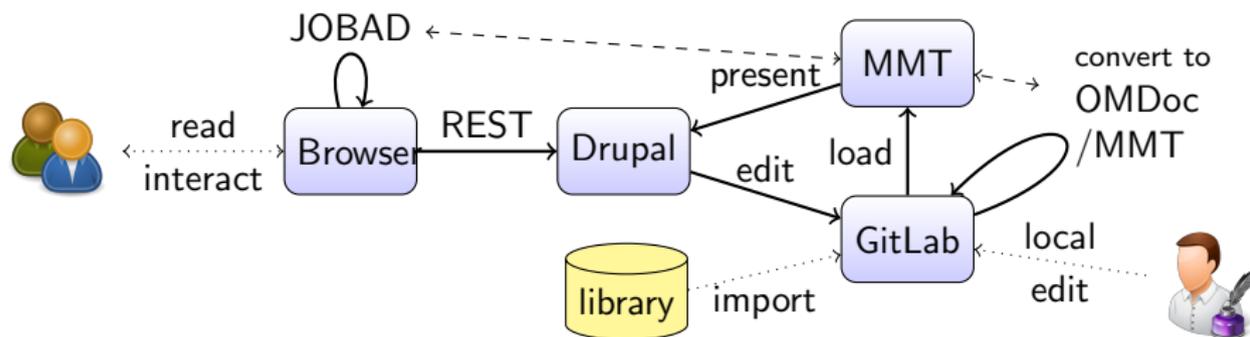
- ▶ MathHub must support
 - ▶ unique and representative user accounts as math developer identities
 - ▶ access rights management (private repositories/public escrow)
 - ▶ overlay periodicals (notification of interesting developments)
 - ▶ tagging/labeling/karma
 - ▶ documentation/help/forums/faqs, . . .

Best-of-Breed Role Model: Drupal: open source container user/management system

- ▶ ▶ well-maintained core
- ▶ ▶ extremely versatile extension system ($\geq 30k$ community add-ons)

MathHub Architecture

- ▶ **MathHub:** Three core components (meet requirements above)
 - ▶ **Representation:** *OMDoc*/MMT mechanized by the MMT system.
 - ▶ **Repositories:** GitLab (git-based public/private repositories)
 - ▶ **Community Support:** Drupal (extensible user/container management)



Additional service providers:

- ▶ JOBAD for embedding services into document
- ▶ importers, e.g. \LaTeX ML for converting \TeX to *OMDoc*, mizar importer
- ▶ MathWebSearch for formula/text search

Common Legal Framework for MathHub: Licensing

- ▶ **Requirement:** a common licensing framework for MathHub (future-proofing)
- ▶ **Problem:** Tension between legitimate goals
 - ▶ users want to start developing content in private (bashful)
 - ▶ community wants a public resource (math content commons)
- ▶ **Idea:** users place their content under **escrow** in MathHub (for safekeeping)
content reverts to an open license after a given time.
- ▶ **Definition 4.1** A license agreement is a **public escrow license**, iff it reverts into a public license (the **target license**) after a specified fixed date (the **escrow date**).
- ▶ **Free Parameters:** target license (e.g. CC-BY) and provisions before escrow date.
- ▶ **Implementation in MathHub:**
 - ▶ users retain copyright on their contributions (of course)
 - ▶ MathHub generates a public escrow license with fixed target date
 - ▶ target date can be extended by three months by user at any time.
 - ▶ repository made public when escrow expires (but user can withdraw content first)
 - ▶ user notification of escrow pending escrow expiry (build trust)

5 Current Projects

Exports from Proof Assistants

- ▶ Export library as MMT projects, store in MathHub
- ▶ Written as part of proof assistant (difficult)
- ▶ OAF chooses representative initial selection
 - ▶ Mizar: set theoretical (initial export done (with Josef Urban))
 - ▶ HOL Light: higher-order logic (initial export done (Rabe/Kaliszyk))
 - ▶ Coq or Matita: type theoretical
 - ▶ IMPS: heterogeneous method
 - ▶ PVS: rich foundational language

Example feature: Search in the MMT API/MathHub

Enter [Java regular expressions](#) to filter based on the URI of a declaration

Namespace	<input type="text"/>
Theory	<input type="text"/>
Name	<input type="text"/>

Enter an expression over theory

Use $\$x,y,z$: to enter unification variables.

Search

type of **MOD_EQ**

$\vdash \forall m:\text{num} . \forall n:\text{num} . \forall p:\text{num} . \forall q:\text{num} . m = n + q * p \implies m \text{ MOD } p = n \text{ MOD } p$

type of **MOD_MULT_ADD**

$\vdash \forall m:\text{num} . \forall n:\text{num} . \forall p:\text{num} . (m * n + p) \text{ MOD } n = p \text{ MOD } n$

Goal: Towards Library Integration

- ▶ Refactor exports to introduce modularity
- ▶ 2 options
 - ▶ systematically during export (e.g., one theory for every HOL type definition)
 - ▶ heuristic or interactive MMT API-based refactoring
- ▶ Collect correspondences between concepts in different libraries
(heuristically or interactively)
- ▶ Relate isomorphic theories across languages
- ▶ Use partial morphisms to translate libraries

Conclusions

- ▶ QED had an influence (at least on me, \rightsquigarrow my students, probably you too)

Conclusions

- ▶ QED had an influence (at least on me, \rightsquigarrow my students, probably you too)
- ▶ **Problem**: interoperability of foundations, logics, and libraries
- ▶ **Solution**: Pluralism at all levels (e.g. via MMT)

Conclusions

- ▶ QED had an influence (at least on me, \leadsto my students, probably you too)
- ▶ **Problem**: interoperability of foundations, logics, and libraries
- ▶ **Solution**: Pluralism at all levels (e.g. via MMT)
- ▶ **Problem**: full formalization is too costly and inflexible (and does not solve many problems)
- ▶ **Solution**: Flexiformality

Conclusions

- ▶ QED had an influence (at least on me, \leadsto my students, probably you too)
- ▶ **Problem**: interoperability of foundations, logics, and libraries
- ▶ **Solution**: Pluralism at all levels (e.g. via MMT)
- ▶ **Problem**: full formalization is too costly and inflexible (and does not solve many problems)
- ▶ **Solution**: Flexiformality
- ▶ **Problem**: Need a community effort to realize QED
- ▶ **Solution**: MathHub.info

Conclusions

- ▶ QED had an influence (at least on me, \rightsquigarrow my students, probably you too)
- ▶ **Problem:** interoperability of foundations, logics, and libraries
- ▶ **Solution:** Pluralism at all levels (e.g. via MMT)
- ▶ **Problem:** full formalization is too costly and inflexible (and does not solve many problems)
- ▶ **Solution:** Flexiformality
- ▶ **Problem:** Need a community effort to realize QED
- ▶ **Solution:** MathHub.info
- ▶ **Upon Re-Reading:** Peter Andrews saw all of this 20 years ago. (one of my heros)

 Henk Barendregt and Arjeh M. Cohen.

Electronic communication of mathematics and the interaction of computer algebra systems and proof assistants.

Journal of Symbolic Computation, 32:3–22, 2001.

 Michael Kohlhase.

The flexiformalist manifesto.

In Andrei Voronkov, Viorel Negru, Tetsuo Ida, Tudor Jebelean, Dana Petcu, Stephen M. Watt, and Daniela Zaharie, editors, *14th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012)*, pages 30–36, Timisoara, Romania, 2013. IEEE Press.

 F. Rabe and C. Schürmann.

A Practical Module System for LF.

In J. Cheney and A. Felty, editors, *Proceedings of the Workshop on Logical Frameworks: Meta-Theory and Practice (LFMTP)*, volume LFMTP'09 of *ACM International Conference Proceeding Series*, pages 40–48. ACM Press, 2009.

 Freek Wiedijk.

The “de Bruijn factor”.

web page at <http://www.cs.ru.nl/~freek/factor/>, 2012.



Wordnet: A lexical database for english.

project page at <https://wordnet.princeton.edu/>.

last 5. 6. 2013.