OAF: An Open Archive of Formalizations

Mihnea Iancu

Jacobs University Bremen

April 12, 2013

Motivation

- numerous deduction systems exist
- most have associated formal libraries
- many notions and results formalized in several libraries
- individual libraries already too large to oversee
- need for integration and management

OAFF : Goals

- an Open Archive of (Flexi)Formalizations
- universal archiving solution for formal math libraries
  - generic wrt. logics and implementations
  - aware of the semantics

  to provide meaningful services
- content integration (for formal libraries)
- active documents

  commenting/rating/refereeing
- dissemination channels

  aggregate journals of formalizations

OAF Architecture

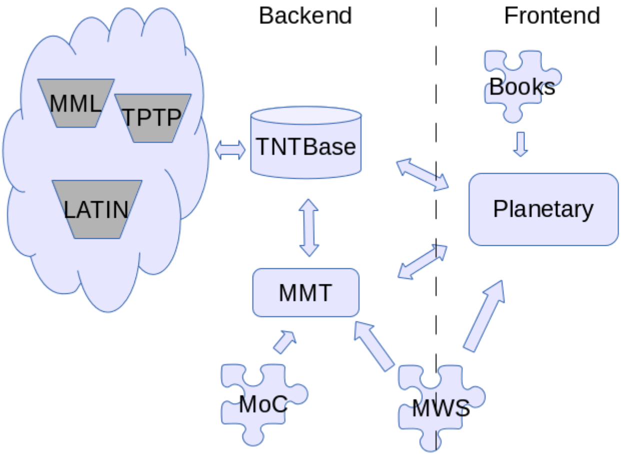
- Backend
  - MMT Kernel

    LATIN logic graph
  - Database

    versioned XML database (TNTBase)
- Frontend
  - Drupal content management system

    Planetary framework

OAF Architecture



Current State

Archives

- around 50GB of data
- urtheories, LATIN, MML, TPTP, OpenMath

Services/Applications

- change management via MMT
- interactive browsing via MMT/JOBAD
- search via MathWebSearch
- discussion forums, local comments via Drupal/Planetary

Current State

Archives

- around 50GB of data
- urtheories, LATIN, MML, TPTP, OpenMath

Services/Applications

- change management via MMT
- interactive browsing via MMT/JOBAD
- search via MathWebSearch
- discussion forums, local comments via Drupal/Planetary

Integration between MMT and Planetary still limited

Archives (Mizar Mathematical Library)

Mizar Project

- started around 1973
- based on a variant of first order logic
- design goal was to be simultaneously readable by mathematicians and verifiable by computer software
- complex statement level declarations

  definition, theorem, lemma, scheme, registration, notation, reservation

Mizar Mathematical Library

- based on build-in Mizar notions and Tarski-Grothendieck set theory
- contains articles formally verified by the Mizar system
- articles are collected and organized by a library committee
- latest version (4.181.1147) includes:
  - 1150 articles written by 241 authors
  - 51762 theorems, 10158 definitions, 787 schemes, 11008 registrations, 7501 symbols.

MML Import

- based on Josef Urban's XML export

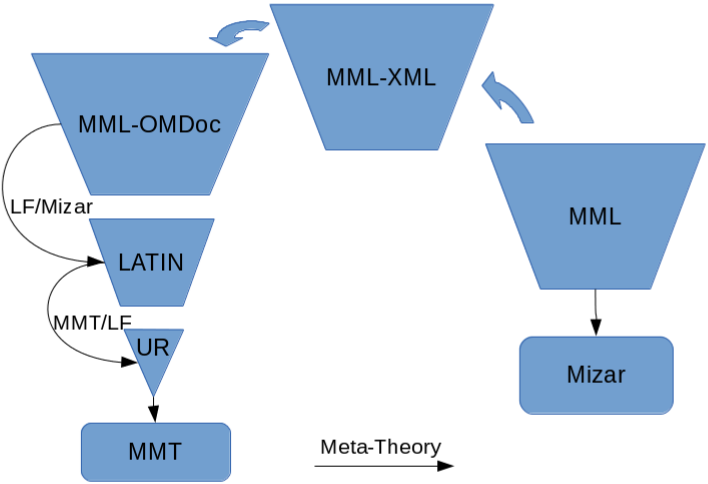
  based on constructor level language

- uses declaration patterns to preserve structure of Mizar statements

  e.g definition, theorem, scheme

- grounded on formalization of Mizar logic in LF
- currently have two versions (1132 and 1147)
- can use OMDoc-based services for (OMDoc)MML

MML Import



MML-XML
MML-OMDoc
MML
Mizar
LF/Miz ar
LATIN
MMT/LF
UR
MMT
Meta-Theory

Services (Search and Change Management)

Search

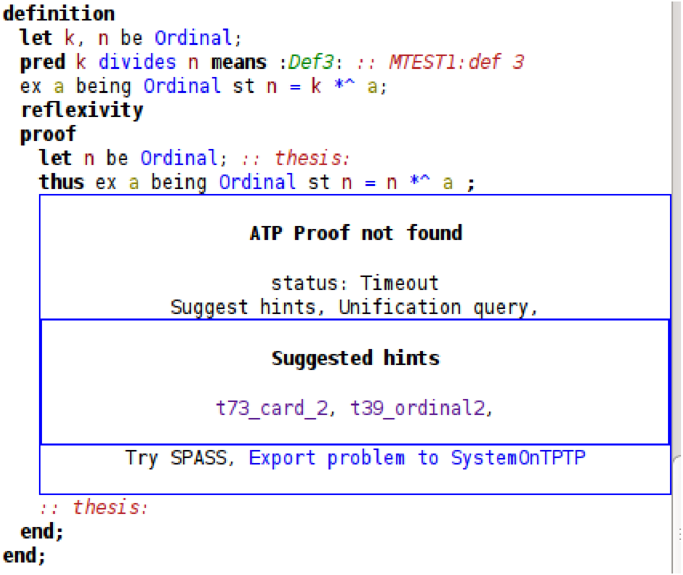
- MathWebSearch (MWS) formula search engine
  - crawler subsystems

    collect data from archives, convert to MWS harvests
  - core system

    builds search index from harvests, processes queries
  - RESTful interface

    HTTP API for interacting with the system
- well integrated with MMT

Search in Mizar



```
definition
 let k, n be Ordinal;
 pred k divides n means :Def3: :: MTEST1:def 3
 ex a being Ordinal st n = k *^ a;
 reflexivity
 proof
  let n be Ordinal; :: thesis:
  thus ex a being Ordinal st n = n *^ a ;

                    ATP Proof not found

                     status: Timeout
           Suggest hints, Unification query,

                     Suggested hints

               t73_card_2, t39_ordinal2,

       Try SPASS, Export problem to SystemOnTPTP

  :: thesis:
 end;
end;
```

MoC in MMT

Management of Change

- MoC usually involves
  - detect changes

    see if/how something changed
  - compute affected items

    maintain some notion of dependency
  - handle/identify conflicts

    in SE typically re-compile e.g. Eclipse

MoC in MMT

Management of Change

- MoC usually involves
  - detect changes

    see if/how something changed
  - compute affected items

    maintain some notion of dependency
  - handle/identify conflicts

    in SE typically re-compile e.g. Eclipse

Goals of MMT MoC

- semantic differencing
- fine-grained dependencies
- impact propagation
- some form of a validity guarantee

MMT Example

MMT Notions

theories contain constant declarations

constants have components (type and definiens)

components represented as MMT/OPENMATH terms

URIs for each theory/constant/component

Rev$_1$

$$
\begin{array}{l}
PL = \{ \\
\quad bool : \mathtt{type} \\
\quad \Rightarrow : bool \to bool \to bool \\
\quad \wedge : bool \to bool \to bool \\
\quad \Leftrightarrow : bool \to bool \to bool \\
\qquad = \lambda x.\lambda y.(x \Rightarrow y) \wedge (y \Rightarrow x) \\
\}
\end{array}
$$

Rev$_2$

$$
\begin{array}{l}
PL = \{ \\
\quad form : \mathtt{type} \\
\quad \neg : form \to form \\
\quad \wedge : bool \to bool \to bool \\
\quad \Leftrightarrow : bool \to bool \to bool \\
\qquad = \lambda x.\lambda y.(x \Rightarrow y) \wedge (y \Rightarrow x) \\
\}
\end{array}
$$

MMT Example

MMT Notions

theories	contain constant declarations
constants	have components (type and definiens)
components	represented as MMT/OPENMATH terms
URIs	for each theory/constant/component

Rev_1

$$
\begin{aligned}
&PL = \{\\
&\quad bool : \mathtt{type}\\
&\quad \Rightarrow : bool \to bool \to bool\\
&\quad \wedge : bool \to bool \to bool\\
&\quad \Leftrightarrow : bool \to bool \to bool\\
&\qquad = \lambda x.\lambda y.(x \Rightarrow y) \wedge (y \Rightarrow x)\\
&\}
\end{aligned}
$$

Rev_2

$$
\begin{aligned}
&PL = \{\\
&\quad form : \mathtt{type}\\
&\quad \neg : form \to form\\
&\quad \wedge : bool \to bool \to bool\\
&\quad \Leftrightarrow : bool \to bool \to bool\\
&\qquad = \lambda x.\lambda y.(x \Rightarrow y) \wedge (y \Rightarrow x)\\
&\}
\end{aligned}
$$

Semantic Differencing

- we extend $\textrm{M}\textsc{mt}$ with a language of (strict) changes
  - add ($\mathcal{A}$) and delete ($\mathcal{D}$) constants
  - update ($\mathcal{U}$) components

Diff	Δ	$::=$	$\cdot \mid \Delta, \delta$
Change	δ	$::=$	$\mathcal{A}(T, c : \omega = \omega') \mid \mathcal{D}(T, c : \omega = \omega') \mid$
			$\mathcal{U}(T, c, o, \omega, \omega')$
Component	o	$::=$	$\texttt{tp} \mid \texttt{def}$
Box Terms	ω	$::=$	$\boxed{\omega} \mid \boxed{\cdot}^{*}$

Pragmatic Changes

- composition of strict ones

  e.g. rename as pair of add and delete

- carry impact semantics

  e.g. for a rename update references

- defined computationally (implemented)

  when can they be constructed, what is their change propagation

- constructed in a separate refinement step

Fine-grained dependencies

- in MMT, validation units are individual components (types and definiens)
- we distinguish two types of dependencies
  - syntactic dependencies
    - declaration level
    - foundation-independent
    - occurs-in relation
  - semantic dependencies
    - component level
    - foundation-dependent
    - trace lookups during foundational validation

Fine-grained dependencies

- in MMT, validation units are individual components (types and definiens)
- we distinguish two types of dependencies
  - syntactic dependencies
    - declaration level
    - foundation-independent
    - occurs-in relation
  - semantic dependencies
    - component level
    - foundation-dependent
    - trace lookups during foundational validation
- dependencies are indexed by MMT and are available at any time

Impact Propagation

- key idea : propagation as diff enrichment process
- impact propagation of a diff Δ is another diff $\overline{\Delta}$ that :
  - marks impacted components

    by surrounding with OpenMath error terms
  - automatically propagates pragmatic changes

    using impact semantics

Impact Propagation

- key idea : propagation as diff enrichment process
- impact propagation of a diff Δ is another diff $\overline{\Delta}$ that :
    - marks impacted components

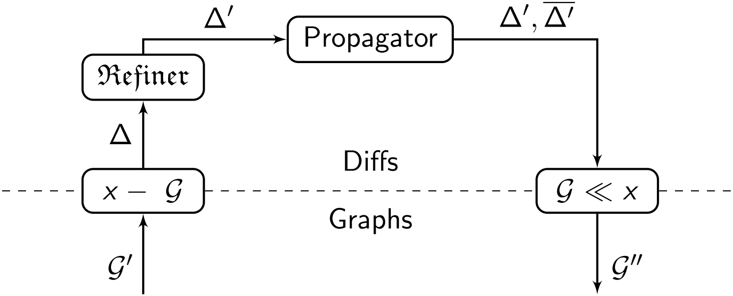
      by surrounding with OpenMath error terms
    - automatically propagates pragmatic changes

      using impact semantics

Theorem

After all error terms are replaced with valid terms in $\mathcal{G} \ll \Delta \ll \overline{\Delta}$, the resulting theory graph is valid.

Workflow Example (relative to a graph $\mathcal{G}$)



Conclusion

- Open Archive of Formalizations
- integrate of formal libraries

  share existing knowledge, make it available to new systems
- provide MKM services

  change management, search, presentation, forum-based discussions
- separation of concerns

  knowledge formalization vs service/application development