

Logosphere

A Digital Library of Formal Proof Content + Meaning

Carsten Schürmann

Yale University

October 2003

Motivation

We have many libraries

- full of mathematical knowledge,
- formalized in many incompatible systems,
- basically without any sharing,
- that are (often) inaccessible.

Motivation (cont'd)

We observe that

- formal methods are more and more used in industry,
- the market is seriously fragmented,
- industry depends more and more on one single system.

Therefore, the time has come
to control logical differences in libraries.

Motivation (cont'd)

Requirements posed to the Logosphere project:

- Trustworthy libraries:
NASA needs to verify software and hardware on spacecraft with respect to various libraries.
- Library reuse:
AMD that is using ACL2 wants access to HOL floating point.
- Platform independent libraries:
New hardware manufacturer wants formal methods without committing.

Related Work

FDL library (Cornell, CalTech, Wyoming)

- Similar goals.
- Nuprl based.

OMDoc (CMU, Bremen)

- Abstract syntax for mathematical knowledge.

Omega (Saarbrücken)

- Blackboard architecture.

... (many more)

Vision behind Logosphere

- Sharing *semantic* mathematical knowledge.
- Brings together communities/philosophies.
- Provides access tools.
- Supports of decision procedures.
- Purposely open architecture.
- Soundness guarantees.
- "Plug'n Play" and not "let's start from scratch".
- Goal: Usability.
- Should be attractive to industrial applications.

Outline of the remaining talk

- Logosphere's architecture.
- Inside Logosphere.
- Logosphere's shell.
- Outside Logosphere.
- Conclusion.

Logosphere's architecture

Metaphor from chemistry:

- Put Logosphere into solution “mathematics.”
- Formalized mathematics crystallizes and organizes.



Logosphere's architecture (cont'd)

What's in the solution?

- PVS (yes)
- HOL (very likely)
- Nuprl (hopefully)
- ACL2 (hopefully)
- SAL (likely)
- TPS (hopefully)
- Mizar (possibly)
- Omdoc (yes)

Inside Logosphere

- Foundationally uncommitted logical framework.
- Not just syntax but "semantic" validation.
- Encoding of the various underlying logics: $[\cdot]$

PVS:

$$\mathcal{D} :: F_1, F_2 \vdash F$$

Logosphere:

$$u_1 : hyp [F_1], u_2 : hyp [F_2] \vdash [\mathcal{D}] : pvs [F]$$

Inside Logosphere (cont'd)

- Semantics captured by axiom and inference systems.
- Further example type systems.
- Design of constraint domains?
- Good starting point:

The logical framework LF.

[Frank's talk]

- Dependently typed.
- Tons of experience.
- Stable implementation.

Inside Logosphere (cont'd)

Ideas for Logosphere that won't work.

- Quest for the universal logic.
- “Let's just hack something up.”
- Syntax is semantics.

Inside Logosphere (cont'd)

Semantic exchange of mathematical knowledge.

- Multi platform environment.
- Central database server.
- Partial translations:
 - Proof translation in between systems.
 - Creation and maintenance of translations.
 - Storage and lookup of translations.
- Total translation (Embeddings).
- Shared core? Numbers? Rings? Algebra?

Inside Logosphere (cont'd)

Further functionality:

- Check in proofs developed in one system, check out in another.
- Merge Libraries.
 - Example: PVS and HOL's library of analysis.
 - Library subsumption.

Shell of Logosphere

- Browsing by name. *Knaster-Tarski.*
- By form. *Let L be a complete lattice and let $f : L \rightarrow L$ be an order-preserving function. Then the set of fixed points of f in L is also a complete lattice.*
- By subsumption. *Let $f : \mathcal{P}(N) \rightarrow \mathcal{P}(N)$ be a continuous function. Then f has a least fixed point.*
- By proof. *Give me all proofs that have implicitly proven a form of Knaster-Tarski.*

Shell of Logosphere (cont'd)

Application Interface

- Theorem annotations.
 - Intended use of a theorem.
 - Helper theorems.
 - Proofless conjectures (clearly marked).
- Style checker.

Shell of Logosphere (cont'd)

Abstract Syntax

- Logosphere talks XML: $[[\cdot]]$

$$u_1 : hyp [F_1], u_2 : hyp [F_2] \vdash [D] : pvs [F]$$

```
<logosphere>
  <context>
    <dec> <id>u1</id> <app> <id>hyp</id>  $[[F_1]]$  </app></dec>
    <dec> <id>u2</id> <app> <id>hyp</id>  $[[F_2]]$  </app></dec>
  </context>
   $[[D]]$  <app> <id>pvs</id>  $[[F]]$ </app>
</logosphere>
```

- Proof compactification.

Outside Logosphere

Crystallizing *corresponds to*:

- Populating Logosphere with knowledge.
- Hookup logic the Logosphere.
 - Nuprl (crystallizes!).
 - PVS (crystallizes?) comes with: Analysis, bit vectors, prelude, finite sets, graph theory, automaton theory, little engines of proofs.
 - HOL (crystallizes!) Analysis libraries.
 - TPS (crystallizes!) Library beneficiary.

Outside Logosphere (cont'd)

Crystallizing *consists of*:

- Formal semantic foundation, e.g. proof theory.
- Logosphere Semantic Definition File.
(Axioms, Rules , Semantic annotations)
- System must speak Logosphere.
- "Proof object generation": Benefit:
 - Proof-directed tool debugging.
 - Enhances Reliability.

Outside Logosphere (cont'd)

Sample session: Crystallizing PVS

- Low level proof representation.
- Proof objects.
- Expansion of tactics.
- Level of abstraction?
- Partial translations from and to systems already crystallized with Logosphere.

Outside Logosphere (cont'd)

Sample session: Crystallizing OMDoc

- Large collection of mathematical knowledge.
- Interfaces to many systems.
- File format purely syntactical.
- Add semantics to theorems and proofs.

Conclusion

Logosphere = Content + Meaning

<http://www.logosphere.org>

Let's hope that many crystallize.

A few challenges:

- Transforming tactics.
- Proof objects for decision procedures.
- Navigation within Logosphere.