Hypatheon: A Mathematical Database for PVS Users

Ben L. Di Vito

NASA Langley Research Center
Hampton, Virginia

b.divito@nasa.gov
http://shemesh.larc.nasa.gov/people/bld

6 January 2004

Aerospace Applications of Deduction Technology

NASA Langley has a long history of research in mechanized deduction for verifying dependable computing designs.

- Fault-tolerant computing, safety-critical avionics, etc.
- Recent progress has led to greater use of continuous mathematics
  - Air traffic management (ATM) algorithms
  - Geometric scenarios constrained by the physics of motion
  - Formalization and proof carried out using PVS (SRI International)
  - This is a promising area, producing good results
- Theorem proving tools such as PVS have some great capabilities
  - But they are woefully "under-educated"
- We will need a significant body of formalized mathematical knowledge
  - But formalized math is *very* detailed
- Result: we may need millions of definitions and theorems

Without readily available, large-scale mathematical knowledge, the cost of deductive techniques could limit their uptake by engineers.

It Takes a Village

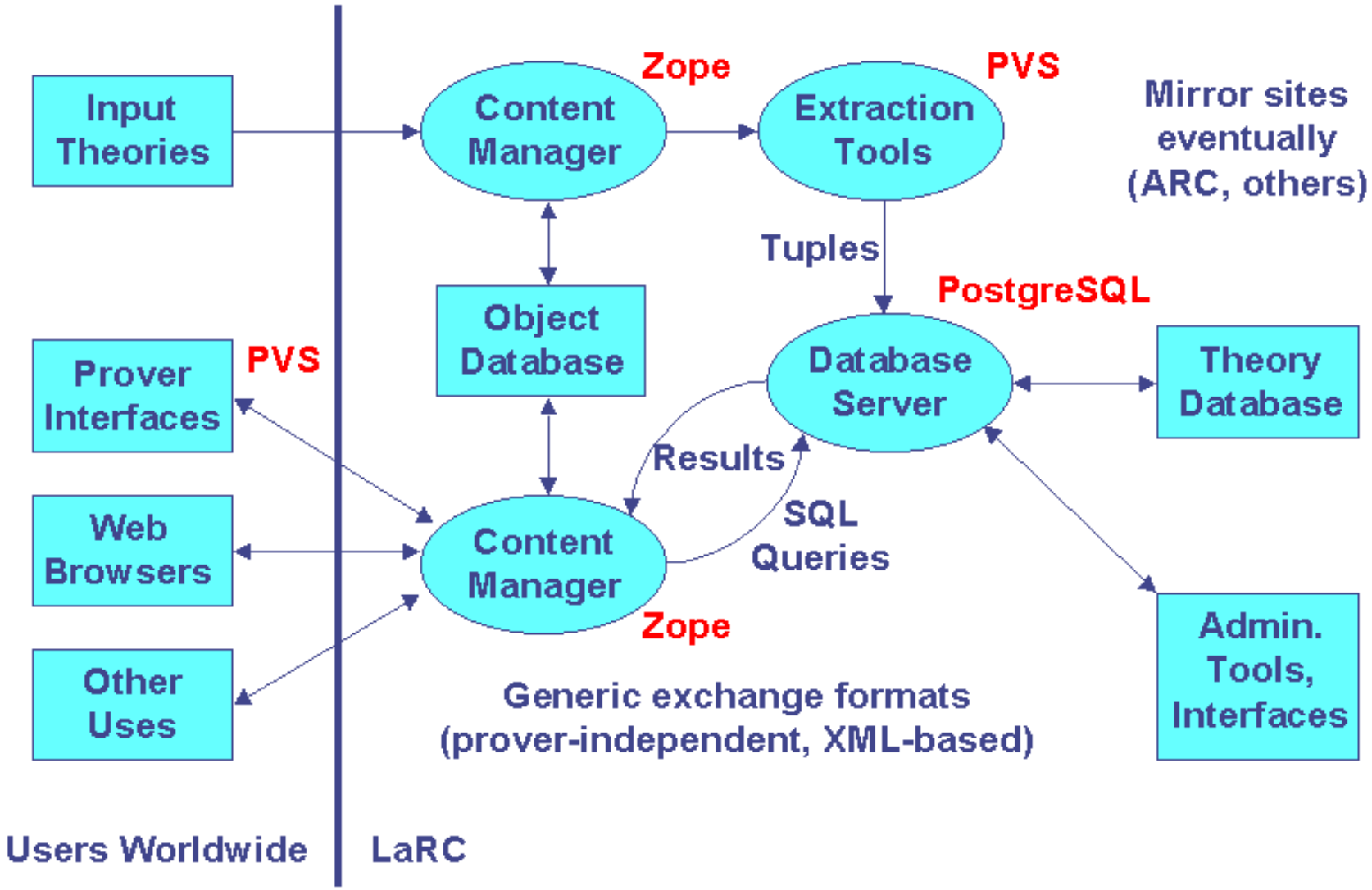
We wish to expand deductive/mathematical knowledge by hosting a dedicated Web server and providing a specialized set of services to PVS users:

- A database of deductive and mathematical artifacts
- A Web-based interface mechanism to:
  - Issue queries against the database
  - Submit new content for inclusion in the database
- A client module to complement PVS
  - Offers proof-side assistance during prover sessions
  - Automates the discovery and acquisition of relevant theorems
- An extensible platform for implementing future services
  - A programmatic interface (API) for invoking services

We provide the service and tools in hopes of attracting contributions from the PVS community

- Users benefit from what we offer
- They are motivated to reciprocate
- A passive collaboration process results

Client-Server Architecture



Populating the Database

- Initial database was created from existing sources of PVS theories:
  - PVS prelude (built-in theories)
  - Libraries distributed with PVS
  - Libraries maintained by NASA Langley (17 so far)
- This has resulted in the following core:
  - 20 libraries
  - 465 theories
  - 1179 functions
  - 3966 formulas
- Later we will add domain-specific formalizations
  - Fault tolerance (by-product of SPIDER project)
  - Air traffic management (various projects)
- We will expand the types of information to be extracted
- Users will be invited and encouraged to submit new content

Current libraries:

prelude
bitvectors
finite_sets
algebra
analysis
arrays
bags
calculus
digraphs
div
fixedpoints
graphs
mod
nat_funs
number_theory
powersets
reals
series
trig
vectors

Hypatheon

| Home | Introduction | Obtaining and Using the PVS client | Submit content | Query the database |

Database of Deductive Knowledge

Welcome to the Hypatheon database of deductive knowledge. Here you will find a collection of mathematics formalized using SRI's PVS language and tools. This database service is provided and maintained by the formal methods team at NASA Langley Research Center. It has been developed under NASA's Engineering for Complex Systems Program.

The following information and services are available:

- Introduction
- Obtaining and Using the PVS client
- Submit content
- Query the database

The Hypatheon development team welcomes your feedback and suggestions.

Curator and Responsible NASA Official: Ben Di Vito
larc privacy statement
last modified: October 22, 2003 3:39 pm GMT-4

Note: The link to external site tag identifies links that are outside of the NASA domain.

Hypatheon

Query the database

The database may be searched directly from the following input forms. Tabular results are returned and displayed by your browser. A PVS client is also available for proof-side searching.

Search for Declarations:

- Search for lemmas that refer to functions
- Search for lemma names by pattern
- Search for function names by pattern
- Search for functions that refer to other functions

Search for Theories:

- Search for theory names by pattern
- Find theories required by given theory
- Find transitive closure of theories that are required by a given theory
- Find theories that depend on given theory
- Find transitive closure of theories that depend on given theory

Search for Libraries:

- List information about libraries

Display database information:

- Show database summary statistics

October 22, 2003 3:48 pm GMT-4

Hypatheon

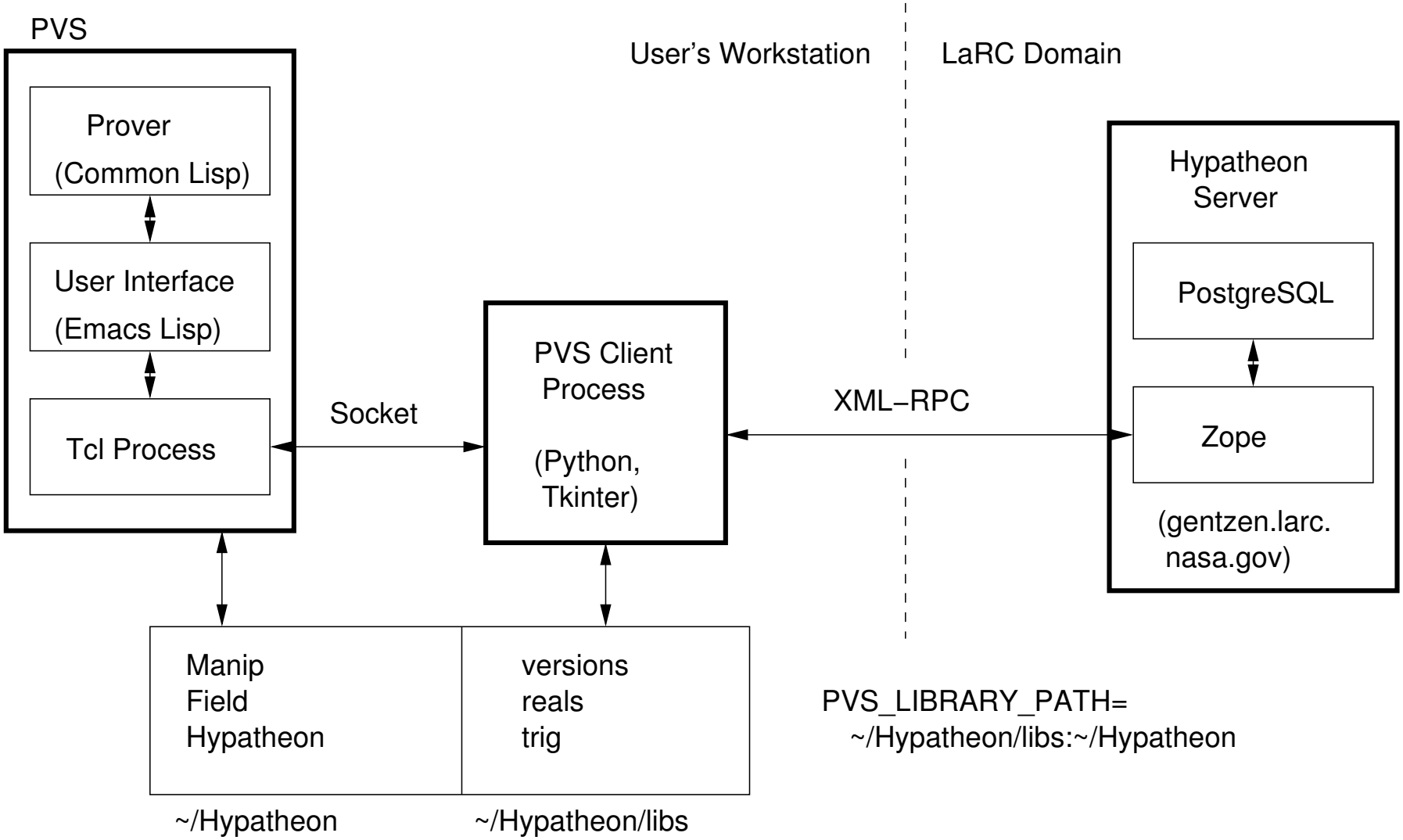
---

13 records found.

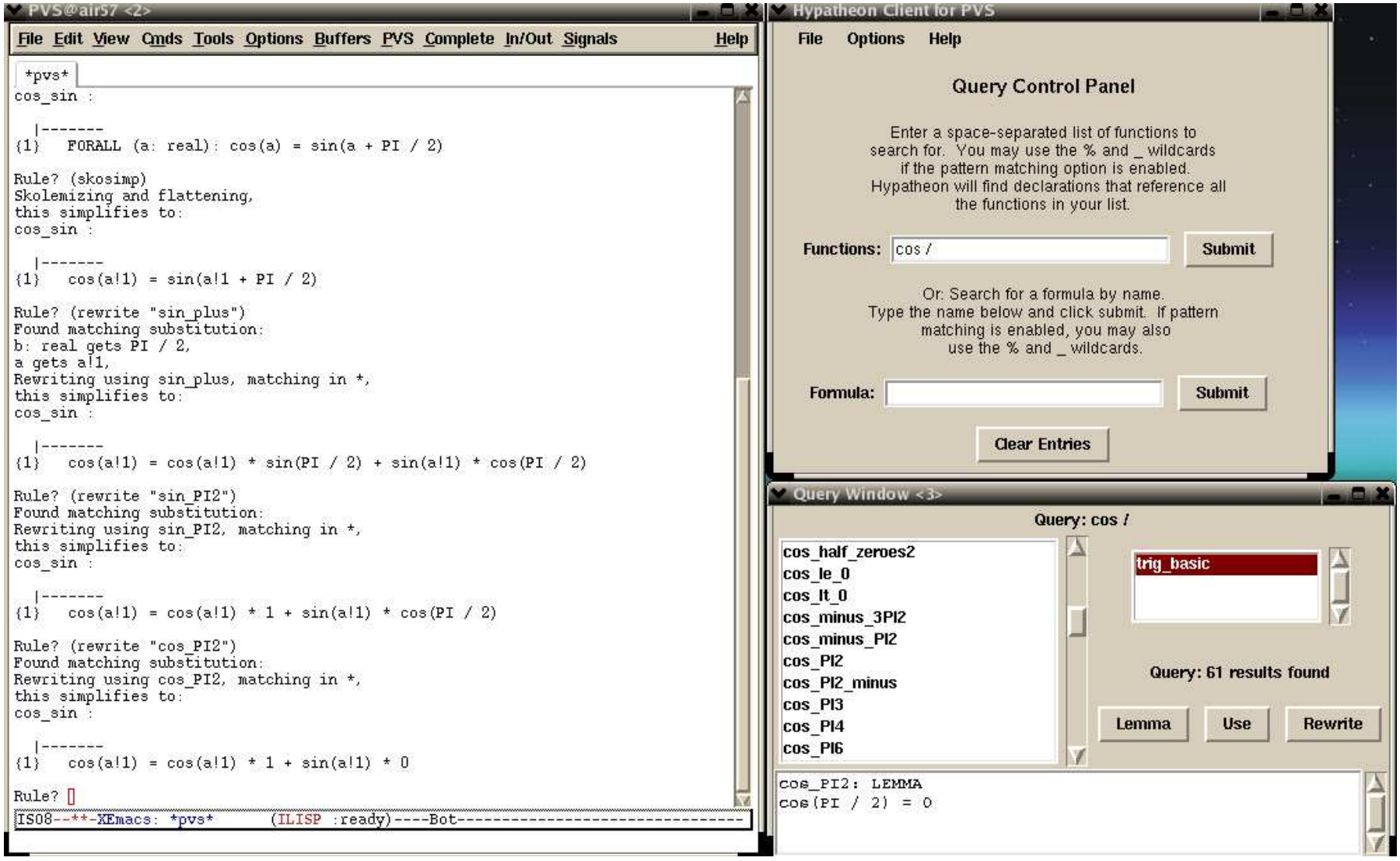
Declaration	Theory	Library
Law_Cosines	law_cosines	trig
sq_dist_is_dist_sq	position	vectors
dist_triangle	position	vectors
law_cosines	law_cos_pos2D	vectors
law_cosines_alt	law_cos_pos2D	vectors
law_cosines_bnd	law_cos_pos2D	vectors
sq_dist_is_dist_sq	position2D	vectors
dist_triangle	position2D	vectors
sq_dist_is_dist_sq	position3D	vectors
dist_triangle	position3D	vectors
law_cosines	law_cos_pos3D	vectors
law_cosines_alt	law_cos_pos3D	vectors
law_cosines_bnd	law_cos_pos3D	vectors

Results produced by Hypatheon on October 22, 2003 3:53 pm GMT-4.

Design of PVS Client



PVS Client Module



Pragmatic Issues

Several aspects of the overall concept/design are still under study:

- Attaching status/maturity indicators to libraries
  - E.g., experimental, stable, deprecated, obsolete
- Coexistence of multiple library versions
  - Incompatible library updates (major versions)
  - Propagation of revisions through database
- Policies on submissions, maintainer responsibilities
- Re-typechecking of libraries by client
- Treatment of strategy packages
- Addition of editor role(s)
  - People who are responsible for content in various areas
  - Decide what to accept, work with submitters/maintainers, etc.

Advanced Queries

Next step in query development aims for greater automation.

- Need heuristics for ranking search results
- Try automatically selecting suitable lemmas
- Imagine a selection function $\boldsymbol{S} : \mathrm{proof\_state} \times \mathrm{database} \longrightarrow \langle \mathrm{lemmas} \rangle$
- Existing library proofs available as a training dataset for $\boldsymbol{S}$
  - Over 3000 lemma invocation sites to draw on
- Goal is to implement heuristics that consistently pick the "correct" lemma or at least rank it highly
- Computing $\boldsymbol{S}$ efficiently is an issue
  - Augment database to ensure adequate performance
  - Build auxiliary database tables with added relationships
  - Precompute (portions of) $\boldsymbol{S}$ as necessary
- Investigate feasibility of finding variable instantiations

Exploiting Proof Information

Having fully mechanical proofs creates new targets of opportunity.

- Add proof artifacts to database
  - Complete PVS proofs
  - Break into steps to uncover structure
  - Relate proof steps to other lemmas/proofs
- Identify patterns and idioms
- Support searches based on proof content
- Enable proof cloning
  - Help users to clone their own proofs
  - Help users to find and clone others' proofs
  - Semi-automatic tailoring (e.g., substitutions: $f \longrightarrow g$)
  - Example: 2D vs. 3D vector theories

Programmatic Interface

A future task is to design and implement a third database interface.

- Provide API for generic database services
- Support arbitrary SQL queries
- Encourage new users and uses
  - Both researchers and advanced practitioners
  - Possibly of interest to non-PVS communities
- Enable custom proof automation for specialized domains
- Potentially useful to mathematical knowledge management (MKM) groups
  - Import/export data to other notations/formalisms

Plans

- Continue to refine current prototype
- Prepare for public server rollout
  - Early 2004
- Conduct performance/capacity testing
- Goal is for server to support:
  - 1 K libraries
  - 10 K theories
  - 100 K function definitions
  - 1 M theorems (formulas)
- Study knowledge organization issues
- Develop advanced query capabilities
- Add proof handling features
- Pursue data mining opportunities