

# Cooperative repositories for formal proofs

A wiki-based solution

Pierre Corbineau and Cezary Kaliszyk

Foundations group, ICIS  
Radboud Universiteit Nijmegen  
The Netherlands

TYPES topical workshop "Math Wiki"  
Edinburgh, 31<sup>st</sup> october-1<sup>st</sup> november 2007

Introduction

Technology

Consistency issues

The ProofWiki prototype

Conclusion

# Why a library for formalized mathematics ?

Proof assistants for:

- ▶ Software and system verification
- ▶ Formalized mathematics

A proof assistant is nothing without a library of *basic* results.  
*'There has to be somebody who already proved this !'*

Different types of libraries:

- ▶ Base for new developments (standard library)
- ▶ Means of publishing new results

# Online Math Libraries

Non-formal examples:

- ▶ Mathworld and many others

Semi-organized collections:

- ▶ Isabelle library (HTML rendering of summary only)
- ▶ Coq (standard library & contrib) (HTML rendered)
- ▶ *documentation generators*

Organized collections:

- ▶ Mizar MML (Much bigger).
- ▶ Corn (also part of Coq contributions)

Searchable databases:

- ▶ HELM

Online systems:

- ▶ Logiweb (online PDF files)



# Why a cooperative environment ?

- ▶ Formalizing mathematics is tedious
- ▶ We need more people involved
- ▶ We need more visibility (general public)
- ▶ Static online contents is not enough
- ▶ A cooperative environment creates a community
- ▶ Support for tutoring new users



# The wiki architecture

Wiki:

- ▶ Online content publishing framework
- ▶ Online content edition system

Provides useful services:

- ▶ History management and (weak) version control
- ▶ Simple hyperlinks & math rendering ( $\text{\LaTeX}$ vc)
- ▶ Discussion threads
- ▶ Reward: instant publication

Clearly successful approach:

- ▶ Wikipedia, Wiktionary
- ▶ Specialized wikis for many software projects
- ▶ Wikis for research websites ...



# Why a web interface for a proof assistant ?

Proof assistants are:

- ▶ Difficult to install
- ▶ Greedy in resource usage

Formal proofs:

- ▶ Hardly self contained
- ▶ Strong operational meaning
  - ▶ *What does this step do ?*
  - ▶ *What are we proving here ?*

An interactive online interface brings:

- ▶ Immediate and easy access
- ▶ Help by observing the proof execution
- ▶ Possibility to modify and experiment
  - ▶ *Formal proving can be fun !*

# Towards a collaborative online repository for formal mathematics

Combine:

- ▶ Community website
- ▶ Open access to formal proofs for the public
- ▶ Visible result for funding agencies
- ▶ Educational projects (undergraduate and master students)
- ▶ *Development*-suite for proofs
- ▶ Reference database (also with informal contents)



Introduction

**Technology**

Consistency issues

The ProofWiki prototype

Conclusion

# The ProofWeb AJAX interface

Developped by C. Kaliszyk.

Supports different proof assistants:

- ▶ Coq, Isabelle, Lego ...

Current use in education:

- ▶ Web Deduction project (RU Nijmegen, VU Amsterdam).

# Embedding into a Wiki framework

Edit mode

writable

flat proof text

special comments

executable proofs

View mode

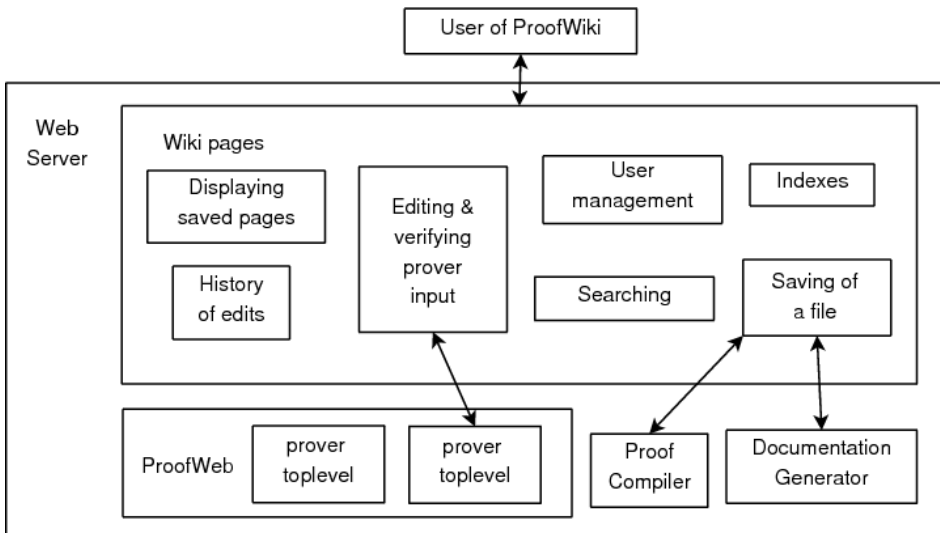
read-only

syntax highlighting, links

HTML documentation

executable proofs

# Architecture



# Practical issues

## Security:

- ▶ Access control policy
- ▶ Arbitrary code execution & DOS attacks

## Solutions used:

- ▶ Sandboxing
- ▶ Limit on session number and timeouts

## Bottleneck:

- ▶ Recompiling and updating dependencies
- ▶ Use of an asynchronous crawler

Introduction

Technology

**Consistency issues**

The ProofWiki prototype

Conclusion

# Formal vs Informal repository

Informal wiki:

- ▶ Dangling references
- ▶ Incomplete articles

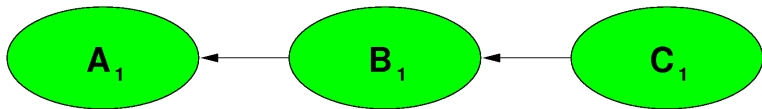
Formal wiki:

- ▶ Keep dependencies as accurate as possible

Three consistency strategies ...

# Dynamic dependencies

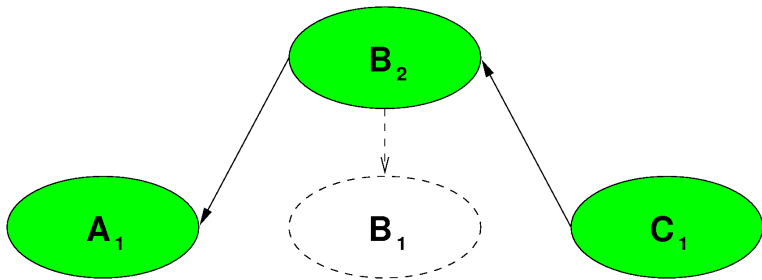
Always depend on the latest version.





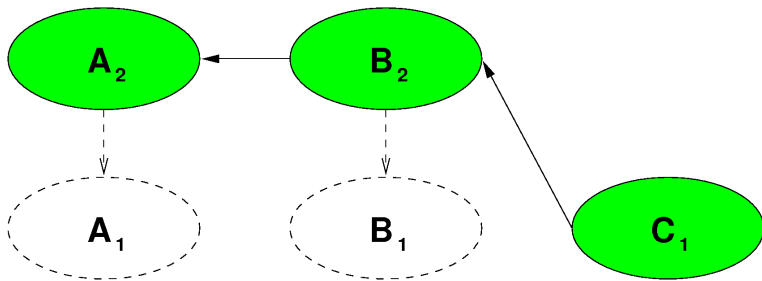
# Dynamic dependencies

Always depend on the latest version.



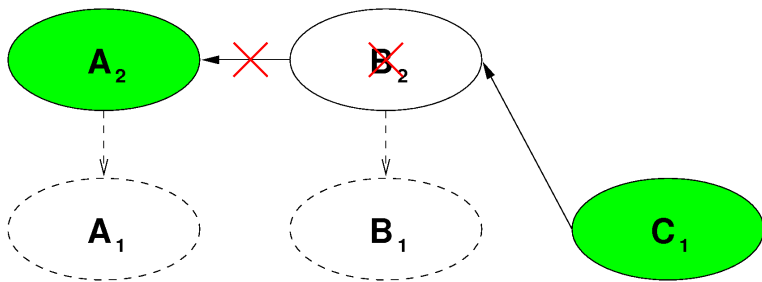
# Dynamic dependencies

Always depend on the latest version.



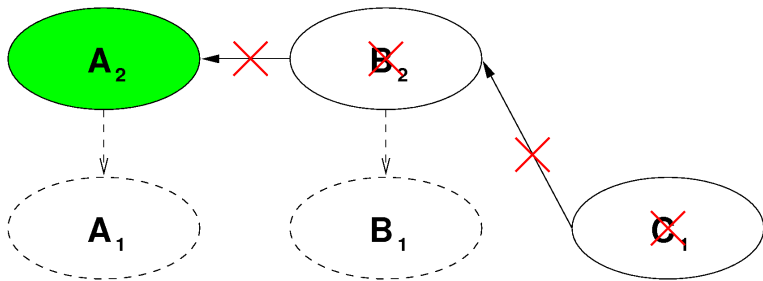
# Dynamic dependencies

Always depend on the latest version.



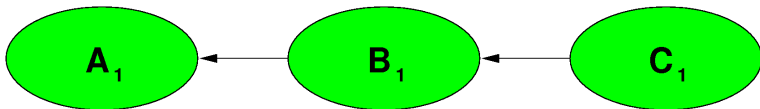
# Dynamic dependencies

Always depend on the latest version.



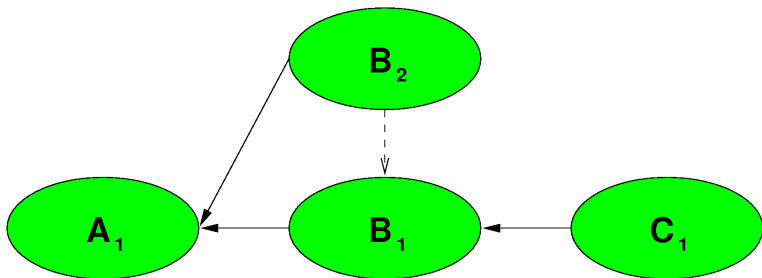
# Static dependencies

No modifications allowed.



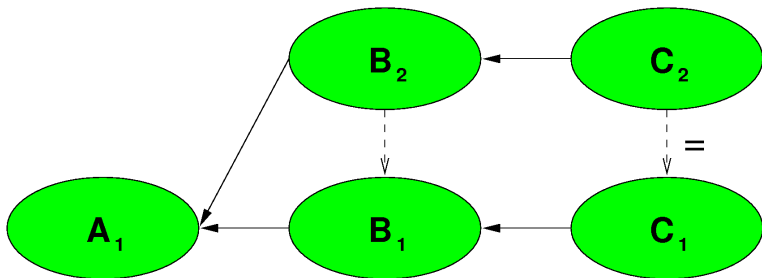
# Static dependencies

No modifications allowed.



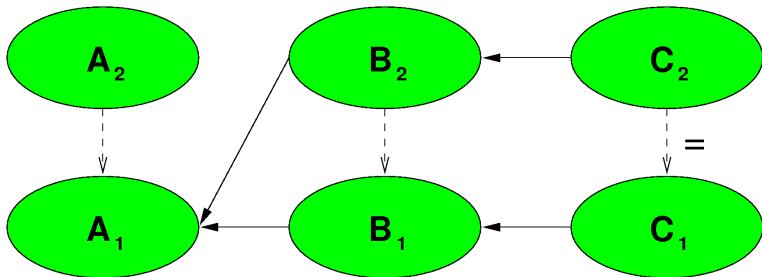
# Static dependencies

No modifications allowed.



# Static dependencies

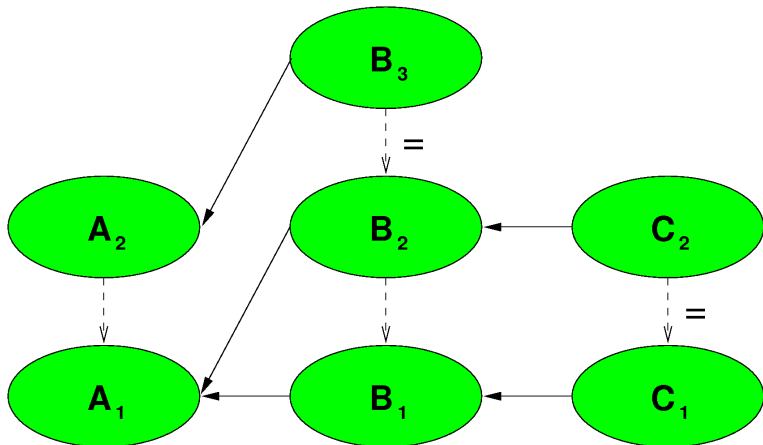
No modifications allowed.





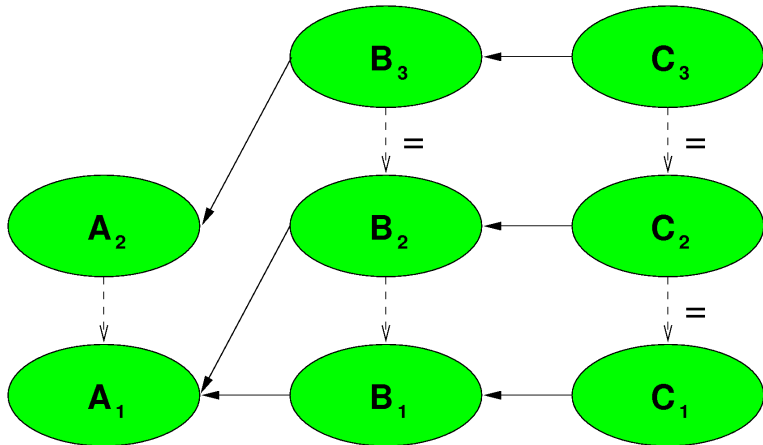
# Static dependencies

No modifications allowed.



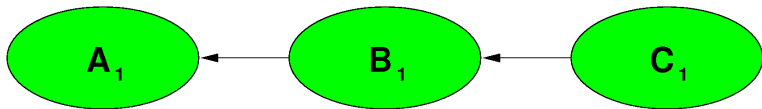
# Static dependencies

No modifications allowed.



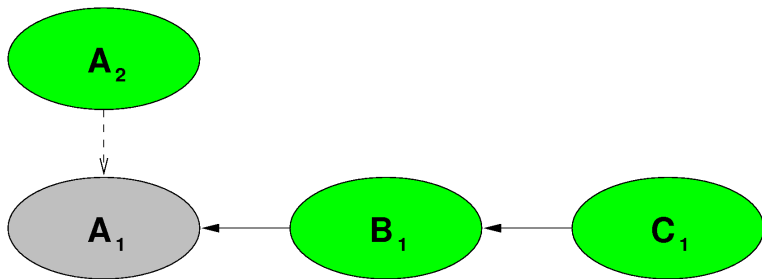
# Colored dependencies

A middle way ?



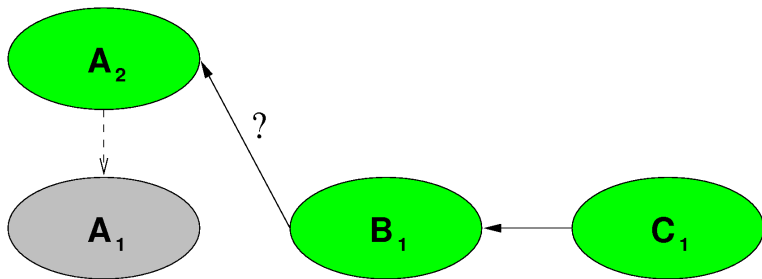
# Colored dependencies

A middle way ?



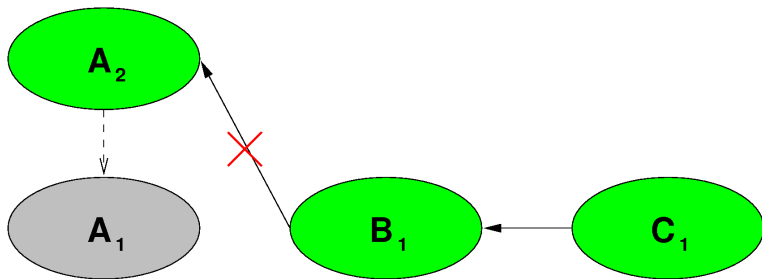
# Colored dependencies

A middle way ?



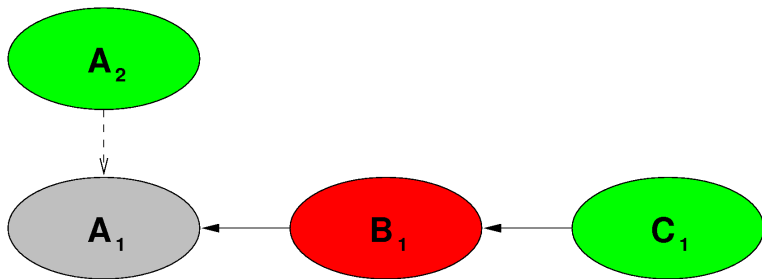
# Colored dependencies

A middle way ?



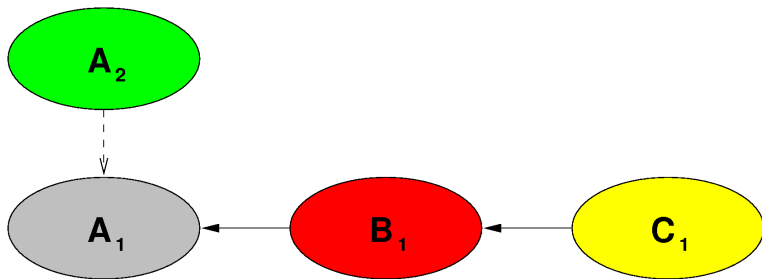
# Colored dependencies

A middle way ?



# Colored dependencies

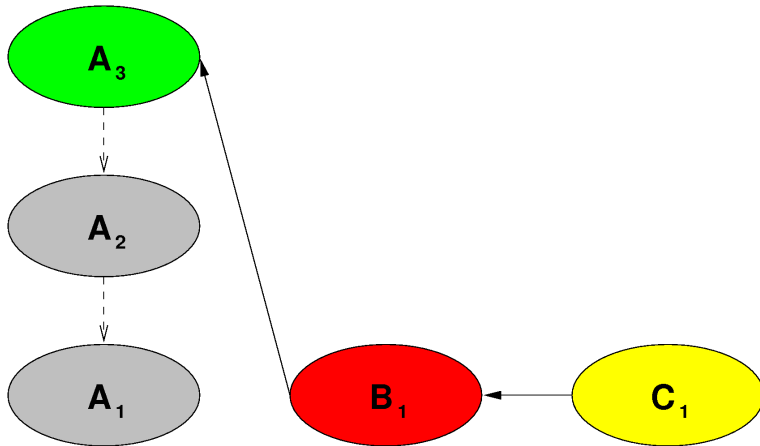
A middle way ?





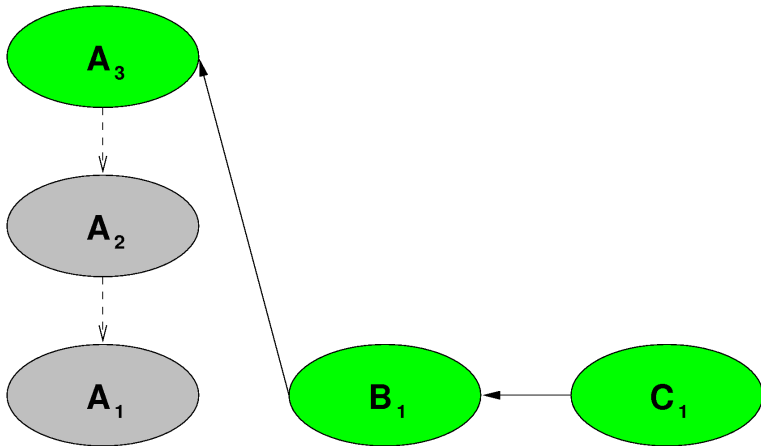
# Colored dependencies

A middle way ?



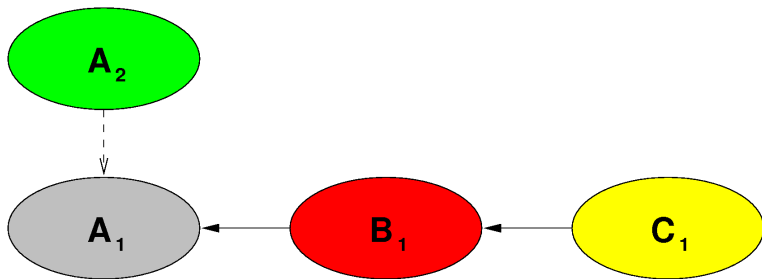
# Colored dependencies

A middle way ?



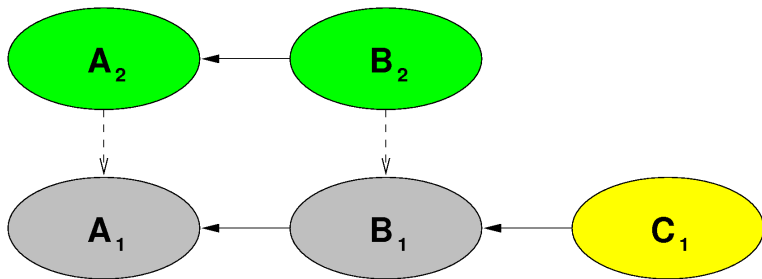
# Colored dependencies

A middle way ?



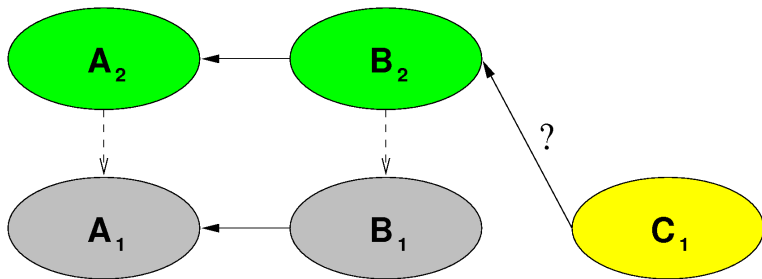
# Colored dependencies

A middle way ?



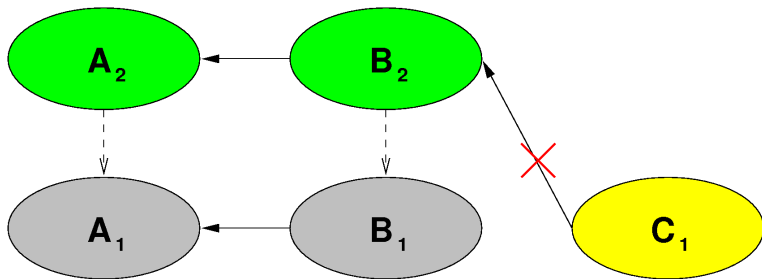
# Colored dependencies

A middle way ?



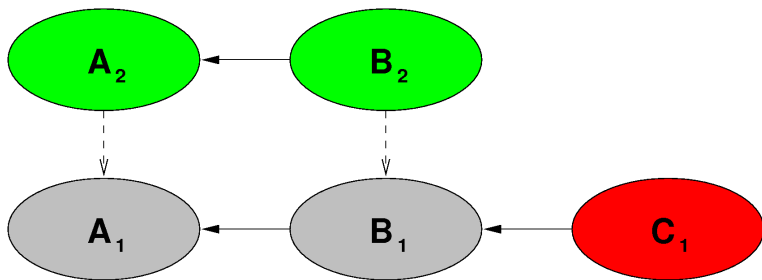
# Colored dependencies

A middle way ?



# Colored dependencies

A middle way ?



Introduction

Technology

Consistency issues

**The ProofWiki prototype**

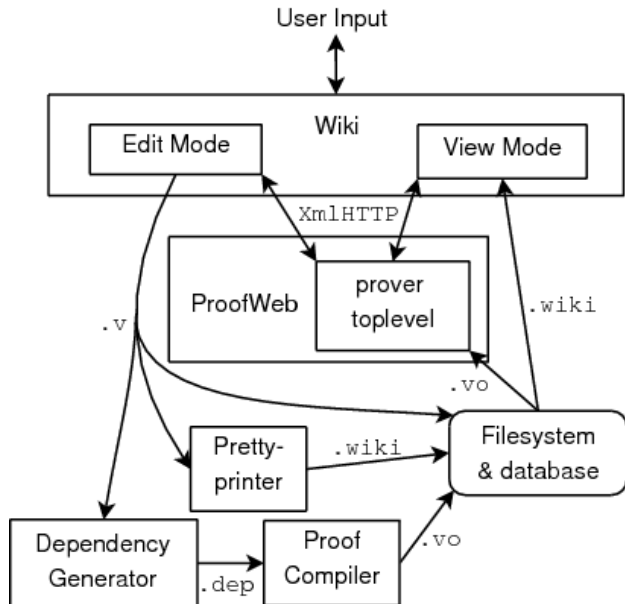
Conclusion



# Off the shelf components

- ▶ Prover: Coq
- ▶ Documentation generator: customized coqdoc
- ▶ Web interface: ProofWeb
- ▶ Wiki Codebase: Mediawiki (PHP-based)

# Dataflow



# Screenshot: Edit Mode

File Edit View History Bookmarks Tools Help

http://hair-dryer.cs.ru.nl/~cek/mediawiki-1.8.2/index.php?title=Wiki.FiniteSums&action=edit

131.174.40.134 talk for this ip log in / create account

article discussion **edit** history

## Editing Wiki.FiniteSums

**Warning:** You are not logged in. Your IP address will be recorded in this page's edit history.

Templates Navigation HELP Coq Documentation

```
Require Import Arith.
Require Import Omega.
Require Import Wiki.Identity.

Fixpoint Sum (f:nat -> nat) (n:nat) : nat :=
match n with
| 0 => 0
| S n' => Sum f n' + f n
end.
(** [(Sum f n)]is defined to be  $\sum_{i=0}^n f$ )

Theorem HighSchool_identity : forall n, Sum id n
intros.
induction n.
simpl; omega.
simpl
```

2 subgoals

=====

Sum id 0 \* 2 = 0 \* 1

subgoal 2 is:

Sum id (S n) \* 2 = S n \* S (S n)

Please note that all contributions to ProofWiki may be edited, altered, or removed by other contributors. If you don't want your writing to be edited mercilessly, then don't submit it here. You are also promising us that you wrote this yourself, or copied it from a public domain or similar free resource (see [Project:Copyrights](#) for details). **DO NOT SUBMIT COPYRIGHTED WORK WITHOUT PERMISSION!**

Summary:

**Save page** Show preview Show changes Cancel | [Editing help](#) (opens in new window)

# Screenshot: View Mode

View History Bookmarks Tools Help

http://hair-dryer.cs.ru.nl/~cek/mediawiki-1.8.2/index.php/Wiki.FiniteSums

131.174.40.134 talk for this ip log in / create account

article discussion edit history

## Wiki.FiniteSums

Templates Navigation HELP Coq Documentation

```
Require Import Arith.
Require Import Omega.
Require Wiki.Identity.

Fixpoint Sum (f:nat -> nat) (n:nat) : nat :=
match n with
| 0 => 0
| S n' => Sum f n' + f n
end.
```

(Sum f n) is defined to be  $\sum_{i=0}^n f(i)$

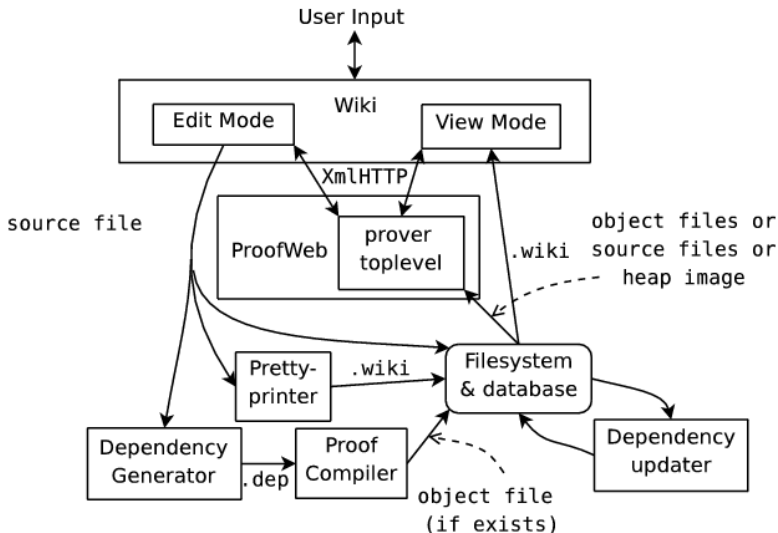
Sum is recursively defined

Search

links here  
d changes



# Towards a more agnostic support of proof assistants



Introduction

Technology

Consistency issues

The ProofWiki prototype

Conclusion

# Future Work

- ▶ *Ad hoc* architecture (easier to manage)
- ▶ More proof assistants
- ▶ Formal / non formal pages (several name spaces)
- ▶ Import / Export feature
- ▶ Implement dependency control
- ▶ Add links from Wikipedia and attract traffic

