Coordinating large formalization projects

Patrick Massot

Université Paris-Saclay

September 14th 2025

Some history

The odd order theorem

Formalized a very well understood result.

The odd order theorem

Formalized a very well understood result.

Small core team around Georges Gonthier.

Before anything like Zulip or GitHub.
Used svn, emails, physical meetings.

The odd order theorem

Formalized a very well understood result.

Small core team around Georges Gonthier.

Before anything like Zulip or GitHub.
Used svn, emails, physical meetings.

Rocq was even called Coq at that time.

Flyspeck (sphere packing in 3D)

Required major refactoring of the paper proof.

Book *Dense Sphere Packings: A Blueprint for Formal Proofs*

Flyspeck (sphere packing in 3D)

Required major refactoring of the paper proof.

Book *Dense Sphere Packings: A Blueprint for Formal Proofs*

Use mostly HOL Light, and some Isabelle/HOL

700 lemmas got cash bounties. About ten people got some.

Tom Hales trained a lot of people, esp. in Vietnam, during the project.

Personal path

- Perfectoid spaces project
- Sphere eversion project
- Liquid tensor experiment
- …

$\rightsquigarrow$ Lean Blueprint

Lean blueprints

Projects using Lean Blueprint

- *Sphere eversion project*
- *Liquid tensor experiment*
- *Unit fractions*
- *Fermat's last theorem for regular primes*
- *Arithmetic Progressions - Almost Periodicity*
- *Polynomial Freiman-Ruzsa Conjecture*
- *Lower bounds for hypothesis testing based on information theory*
- *New Foundations is consistent*
- *Prime number theorem and...*
- *Fermat's Last Theorem for 3*

Projects using Lean Blueprint

- *Fermat's last theorem*
- *Carleson Operators on Doubling Metric Measure Spaces*
- *Infinity Cosmos*
- *Analytic Number Theory Exponent Database*
- *Equational Theories*
- *HoTTLean*
- *Sphere packing in Lean*
- *Localic Caratheodory extensions*
- *Banach Tarski*
- *Bourgain extractor in Lean*
- *Semi circle Law*
- *Seymour*
- *STIR verification*
- *Central limit theorem*

Projects using Lean Blueprint

- *Brownian motion*
- *Formally Verified Arguments of Knowledge*
- *Chandra Furst Lipton*
- *Cambridge combinatorics*
- *Toric varieties in Lean*
- *Iwasaw theory in Lean*
- *Irrationality of $\zeta(3)$*
- *Exceptional set in the abc conjecture*
- *Analysis in Bonn*
- *Extreme Value Distribution Project*
- *Formal proofs from the book*
- *Spectral theorem*

Blueprints

Three main aspects:

- Preparing the mathematical content
- Managing progress and distributing work
- Dynamically refining mathematical content

Lean blueprint

Main principles from the beginning:

- Source should be ordinary TeX files
  - easy to copy–paste from existing sources
  - allow contributions by anyone, using any editor
  - pdf version for free

Lean blueprint

Main principles from the beginning:

- Source should be ordinary TeX files
- Result should include links to Lean declarations (and check they exist)
- Result should make clear what are the next targets
- Get at least a crude approximation of a bilingual text.

Demo

Let’s see what it looks like *in source* and *the result*.

Debate 1: Put everything in one file?

- PROs: no switching, people addicted to copilot feel safer
- CONs: harder for non-formal contributors, editor support is very weak
- used in Prime Number Theorem+ project. A Python script by Ian Jauslin extracts TeX files from Lean files.

See *an example*.

Debate 2: Use `\ref` instead of `\uses`

- Huge loss of flexibility and precisions
- Experience suggests the `\uses` overhead is negligible
- We could still implement warnings when a `\ref` has no `\uses`

Debate 3: Where to put discussions?

- Each item can link to a GitHub issue
- Zulip chat or similar
- We could try a commenting framework like Discourse
- Finding the right balance is tricky

Debate 4: What about huge projects?

Fermat Last Theorem is too big for a single dependency graph.

This was already true for the Liquid tensor experiment (used two graphs).

Better solution is WIP.

Debate 5: Why using LaTeX in 2025?

Debate 5: Why using LaTeX in 2025?

- *Typst* did not exist in 2020

Debate 5: Why using LaTeX in 2025?

- *Typst* did not exist in 2020
- HTML export is not yet good enough, and may never become good enough.

Debate 5: Why using LaTeX in 2025?

- *Typst* did not exist in 2020
- HTML export is not yet good enough, and may never become good enough.
- Most mathematicians still use LaTeX.

LeanBlueprint implementation stack:

- ***plasTeX***: python LaTeX compiler with intermediate representation and plugin system

LeanBlueprint implementation stack:

- ***plasTeX***: python LaTeX compiler with intermediate representation and plugin system
- ***plasTeXdepgraph***: plasTeX plugin, `\uses` macro, draws the graph, infrastructure to customize the graph

LeanBlueprint implementation stack:

- ***plasTeX***: python LaTeX compiler with intermediate representation and plugin system
- ***plasTeXdepgraph***: plasTeX plugin, `\uses` macro, draws the graph, infrastructure to customize the graph
- ***leanblueprint***: plasTeX plugin, `\lean`, `\leanok`, links to Lean declarations. Also contains command line client to setup project (including CI) and compile.

LeanBlueprint implementation stack:

- ***plasTeX***: python LaTeX compiler with intermediate representation and plugin system
- ***plasTeXdepgraph***: plasTeX plugin, `\uses` macro, draws the graph, infrastructure to customize the graph
- ***leanblueprint***: plasTeX plugin, `\lean`, `\leanok`, links to Lean declarations. Also contains command line client to setup project (including CI) and compile.
- ***checkdecl***: tiny Lean library to allowing to check that declarations exist to avoid dead links.

LeanBlueprint implementation stack:

- ***plasTeX***: python LaTeX compiler with intermediate representation and plugin system
- ***plasTeXdepgraph***: plasTeX plugin, `\uses` macro, draws the graph, infrastructure to customize the graph
- ***leanblueprint***: plasTeX plugin, `\lean`, `\leanok`, links to Lean declarations. Also contains command line client to setup project (including CI) and compile.
- ***checkdecl***: tiny Lean library to allowing to check that declarations exist to avoid dead links.

First three items are python+JS. First two have nothing to do with Lean. Third item is Lean-specific but very easy to adapt. Only the fourth item contains Lean code.

Typical project life cycle

Setting up new projects

GitHub template by Pietro Monticone

`leanblueprint` *command line interface*

Project leadership and team

It is crucial to have someone cutting the project in pieces, stating results and reviewing contributions.

Easy proofs are also easy to parallelize and distribute. They play a major role in onboarding and training.

Tracking tools

Popular Zulip threads: outstanding tasks (*example*)

GitHub projects managed by *GitHub workflows*.

Contributing back to Mathlib

The infamous `for_mathlib` *folder* issue.

Upstreaming dashboard can help a bit.

Maintenance after project completion is difficult.

Contributing back to Mathlib

The infamous `for_mathlib` *folder* issue.

Upstreaming dashboard can help a bit.

Maintenance after project completion is difficult.

Key trick: miraculous appearance of *maintainers*.

Future directions

Help from Lean FRO

Lean Focussed Research Organization since July 2023

5-year mission to improve Lean and reach long-term self-sustainability.

Huge work to coordinate Lean and Mathlib modifications.

Reduce technical debt inherited esp. from Lean 3 $\longrightarrow$ Lean 4.

Current relevant FRO work

GitHub actions being worked on (CI, Lean version update...)

Downstream project reports

Future FRO work

From year 3 roadmap:

“**Literate Programming & Next-Generation UX**

Verso will be enhanced to support both PDF extraction and interactive website generation. A new DSL—LaTeX-inspired and tailored for Lean—will be prototyped to support scientific writing.”

Future help from Mathlib Initiative

Announced on July 24th

$5M donated by Alex Gerko for first two years

One goal is to “work on scaling the infrastructure to support an ecosystem of math libraries depending on Mathlib”.