Blueprints for formalisation in Coq

Peter LeFanu Lumsdaine, Stockholm University

I will discuss the use and usefulness of "blueprints" in formalisation, and the design practices they promote; and I will hopefully present a port for Coq of the leanblueprint tool.

Blueprint documents of one sort or another are a venerable tool of software engineering [Bau78]. They can describe many aspects of a software project, but especially often emphasise the dependency structure between components of a system.

As proof assistants have matured, such blueprints (along with other software engineering practices) have been adopted in the formalisation community — especially in larger formalisations, where careful explicit planning is essential, as discussed in [Gon13b]. Again, dependency graphs are a central feature — tracking and surveying dependencies at both the high level of modules and packages, and the more granular level of individual definitions and theorems.

Early in of a project, such a blueprint helps plan towards the overal goal; during the project, it serves to guide work and track progress; and after completion, it becomes part of the documentation, a useful entry-point for readers exploring the development. Besides this, a blueprint can enable and encourage interesting coding techniques, facilitating working not only forwards from basics but also backwards from goal results (which may be initially stated with placeholders for not-yet-defined notions).

In smaller developments, blueprints are more rarely used: one can get by well enough without them, and creating them takes a bit of work upfront — so the path of least resistance is to go without. However, good tooling can change this. The leanblueprint tool[1], by Patrick Massot, is a lightweight plugin for creating such blueprints and integrating them into a Lean development, as seen at for instance [SCM23] and described in [Tao23]. It was written for the Sphere Eversion Project[MND22], and has since been used by many other formalisations in Lean.

Having worked a little with leanblueprint, I found myself sorely missing it when working in Coq; so I have recently (at time of writing) begun a port of it adapted for Coq, which I hope to have ready to present at the workshop.

[1] https://github.com/PatrickMassot/leanblueprint

References

[Bau78] Patrica H. Baucom. "Software Blueprints". In: *Proceedings of the 1978 Annual Conference*. ACM '78. Washington, D.C., USA: Association for Computing Machinery, 1978, pp. 385–392. ISBN: 0897910001. DOI: 10.1145/800127.804131.

[Gon13a] Georges Gonthier. "Engineering mathematics: the odd order theorem proof". In: *SIGPLAN Not.* 48.1 (Jan. 2013), pp. 1–2. ISSN: 0362-1340. DOI: 10.1145/2480359.2429071.

[Gon13b] Georges Gonthier. *Engineering mathematics: the odd order theorem proof (POPL '13 keynote address)*. Video available as supplemental material to [Gon13a]. 2013.

[MND22] Patrick Massot, Oliver Nash, and Floris van Doorn. *The sphere eversion project*. Website. 2022. URL: https://leanprover-community.github.io/sphere-eversion/ (visited on 06/09/2024).

[SCM23] Peter Scholze, Johan Commelin, and Patrick Massot. *Blueprint for the Liquid Tensor Experiment*. Sept. 18, 2023. URL: https://leanprover-community.github.io/liquid/ (visited on 06/09/2024).

[Tao23] Terence Tao. *Formalizing the proof of PFR in Lean4 using Blueprint: a short tour*. Blog post. Nov. 18, 2023. URL: https://terrytao.wordpress.com/2023/11/18/formalizing-the-proof-of-pfr-in-lean4-using-blueprint-a-short-tour/ (visited on 06/09/2024).