A Data-centered User Study for jsCoq

HANNELI C. A. TAVANTE, McGill University

In this talk, we describe an approach for a data-centered user study for proof assistant tools, targeting jsCoq. There is a wide variety of interfaces for Coq (in standalone formats, such as CoqIDE; as web resources, like jsCoq, and plugins for major IDEs). But up to this point, there are no records of user studies involving large amounts of data for none of the existing tools. An analysis centered on user data could improve the overall usability of these interfaces by revealing issues with their design. In the educational field, the investigation could also help lecturers and staff to understand the students' struggles and issues better when learning Coq. This talk shows a work in progress demo for a data collection procedure in jsCoq, and summarizes a few key discussion points for the educators involved in the Coq community.

Additional Key Words and Phrases: Coq, user-study, data collection

1 INTRODUCTION

Multiple studies proposing user interfaces for proof assistants have been published in the last decades. In each edition of *User Interfaces for Theorem Provers* (UITP), it is possible to find a wide range of experiments for both educational and professional purposes.

For the Coq proof assistant, CoqIDE[1] and ProofGeneral [Aspinall 2000] are among the most widely adopted interfaces. More recent attempts consider options with support for Web interfaces, such as jsCoq [Gallego Arias et al. 2017]. The immediate benefit of online environments like jsCoq is that the user does not need to install any tool in their local machine. Hence, they are able to try out the Coq without any setup overhead.

Several projects listed on jsCoq Github page [2] seem to benefit from its portability. In particular, several educational events (Summer/Winter schools, tutorials, workshops) simply embedded the required code to be used on a particular demonstration on jsCoq. For instance, it becomes clear that as an online environment, jsCoq has a lot of potentials to be adopted as an educational resource [Warren et al. 2014].

There are multiple other tools and plugins for Coq (VSCoq [3] for Visual Studio Code, Coqtail for Vim [4] and even a Jupyter Notebook style interface [5]). Other integrated development environments or extensions were proposed (such as CoqPIE [Roe and Smith 2016] and Company-Coq [Pit-Claudel and Courtieu 2016]), broadening options for the end user.

2 USER EXPERIENCE STUDY WITH LARGE DATASETS

Up to this point, there is no record of a large data-centered user study for any of the tools previously mentioned. In prior work, Knobelsdorf et al. [2017] performed surveys with small groups of students to answer the following question: "What kind of problems and issues do students run into when working with Coq, especially usability issues?". Ringer et al. [2020] investigates the development process of users using proof assistants, but it targets experienced users and also reports the results of a small data set. Whereas qualitative data is a perfectly valid approach, it would be interesting to have large datasets to observe how new learners of Coq are performing, and more importantly,

[1]https://coq.inria.fr/refman/practical-tools/coqide.html
[2]https://github.com/jscoq/jscoq
[3]https://github.com/coq-community/vscoq
[4]https://github.com/whonore/Coqtail
[5]https://github.com/EugeneLoy/coq_jupyter

Author's address: Hanneli C. A. Tavante, hanneli.andreazzitavante@mail.mcgill.ca, McGill University.

which types of issues they have when using Coq. An analysis based on big data could also reveal crucial insights for experienced users at the same time.

One way to obtain such a data set would be via jsCoq. Contrary to similar attempts for data massive data collection in IDEs (see the project Blackbox [Brown et al. 2018], for reference), jsCoq is an online tool, and an asynchronous integration with node.js server redirecting the browser data to a database (for example, MongoDB [6]) would be enough to generate data entries. Each record could contain a snapshot of the user text (from the scratchpad), which button they clicked, a cursor to the proof, the output of the Coq worker, and a timestamp. This setting would enable a full reconstruction of the user timeline and their sequence of steps when using jsCoq.

The data collection strategy previously described would also enable institutions to deal with their own privacy restrictions, consent forms, ethics approval, and anonymization mechanisms according to their own norms, instead of relying on a centralized database in an external institution. A similar approach has been described for the LearnOCaml platform ([Canou et al. 2017]) by Ceci et al. [2021].

This data collection followed by data analysis results could bring up multiple other factors related to the overall user experience, not only about the tool interface. For students, it could reveal which topics are harder, point out redundant proof steps and common mistakes. It could also be the key to automated guidance from the platform (for example, automatic suggestion for libraries and refactoring).

3 DISCUSSION AND FUTURE WORK

This presentation aims to show a simple demo of this data collection for jsCoq (still a work in progress during summer 2021), and at the same time invite the community, especially those directly involved with education, to investigate ways to provide students with a better user experience of Coq tools. Analyze the collected data can guide the path for interface enhancements and possibly lower the entry bar to the topic of proof assistants.

A number of related ideas and questions may deserve a place for discussion:

- Is it possible to derive lessons from other existing environments for theorem provers, such as Lean [7]? The Lean community has a detailed introductory tutorial, which seems efficient for both beginners in the field, and also for people with prior experience in proofs, but new to Lean.
- Would it be valid to try a cross-institution effort in the Coq community to improve these tools? The Blackbox project relied on multiple universities for collecting data and analyzing.
- What other research questions could one ask with such a dataset, besides interface and educational-related questions?
- What about the usability of offline tools? Would it make sense to have a Language Server Protocol (LSP) for Coq? There seems to be an open issue on the official project [8].

REFERENCES

David Aspinall. 2000. Proof General: A Generic Tool for Proof Development. In *Tools and Algorithms for Construction and Analysis of Systems, 6th International Conference, TACAS 2000, Held as Part of the European Joint Conferences on the Theory and Practice of Software, ETAPS 2000, Berlin, Germany, March 25 - April 2, 2000, Proceedings (Lecture Notes in Computer Science, Vol. 1785)*, Susanne Graf and Michael I. Schwartzbach (Eds.). Springer, 38–42. https://doi.org/10.1007/3-540-46419-0_3

[6]https://www.mongodb.com/

[7]https://leanprover-community.github.io/index.html

[8]https://github.com/ejgallego/coq-serapi/issues/26

Neil C. C. Brown, Amjad AlTadmri, Sue Sentance, and Michael Kölling. 2018. Blackbox, Five Years On: An Evaluation of a Large-scale Programming Data Collection Project. In *Proceedings of the 2018 ACM Conference on International Computing Education Research, ICER 2018, Espoo, Finland, August 13-15, 2018*, Lauri Malmi, Ari Korhonen, Robert McCartney, and Andrew Petersen (Eds.). ACM, 196–204. https://doi.org/10.1145/3230977.3230991

Benjamin Canou, Roberto Di Cosmo, and Grégoire Henry. 2017. Scaling up functional programming education: under the hood of the OCaml MOOC. *Proc. ACM Program. Lang.* 1, ICFP (2017), 4:1–4:25. https://doi.org/10.1145/3110248

Alana Ceci, Hanneli C. A. Tavante, Brigitte Pientka, and Xujie Si. 2021. Data Collection for the Learn-OCaml Programming Platform: Modelling How Students Develop Typed Functional Programs. In *SIGCSE '21: The 52nd ACM Technical Symposium on Computer Science Education, Virtual Event, USA, March 13-20, 2021*, Mark Sherriff, Laurence D. Merkle, Pamela A. Cutter, Alvaro E. Monge, and Judithe Sheard (Eds.). ACM, 1341. https://doi.org/10.1145/3408877.3439579

Emilio Jesús Gallego Arias, Benoît Pin, and Pierre Jouvelot. 2017. jsCoq: Towards Hybrid Theorem Proving Interfaces. In *Proceedings of the 12th Workshop on User Interfaces for Theorem Provers, Coimbra, Portugal, 2nd July 2016 (Electronic Proceedings in Theoretical Computer Science, Vol. 239)*, Serge Autexier and Pedro Quaresma (Eds.). Open Publishing Association, 15–27. https://doi.org/10.4204/EPTCS.239.2

Maria Knobelsdorf, Christiane Frede, Sebastian Böhne, and Christoph Kreitz. 2017. Theorem Provers as a Learning Tool in Theory of Computation. In *Proceedings of the 2017 ACM Conference on International Computing Education Research, ICER 2017, Tacoma, WA, USA, August 18-20, 2017*, Josh Tenenberg, Donald Chinn, Judy Sheard, and Lauri Malmi (Eds.). ACM, 83–92. https://doi.org/10.1145/3105726.3106184

Clément Pit-Claudel and Pierre Courtieu. 2016. Company-Coq: Taking Proof General one step closer to a real IDE. In *CoqPL'16: The Second International Workshop on Coq for PL.* https://doi.org/10.5281/zenodo.44331

Talia Ringer, Alex Sanchez-Stern, Dan Grossman, and Sorin Lerner. 2020. REPLica: REPL instrumentation for Coq analysis. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, Jasmin Blanchette and Catalin Hritcu (Eds.). ACM, 99–113. https://doi.org/10.1145/3372885.3373823

Kenneth Roe and Scott F. Smith. 2016. CoqPIE: An IDE Aimed at Improving Proof Development Productivity - (Rough Diamond). In *Interactive Theorem Proving - 7th International Conference, ITP 2016, Nancy, France, August 22-25, 2016, Proceedings (Lecture Notes in Computer Science, Vol. 9807)*, Jasmin Christian Blanchette and Stephan Merz (Eds.). Springer, 491–499. https://doi.org/10.1007/978-3-319-43144-4_32

Joe Warren, Scott Rixner, John Greiner, and Stephen Wong. 2014. Facilitating human interaction in an online programming course. In *The 45th ACM Technical Symposium on Computer Science Education, SIGCSE 2014, Atlanta, GA, USA, March 5-8, 2014*, J. D. Dougherty, Kris Nagel, Adrienne Decker, and Kurt Eiselt (Eds.). ACM, 665–670. https://doi.org/10.1145/2538862.2538893