Improving Analysis Tool Documentation for the HSF

Organization: CERN-HSF

Particle physics is an exciting field where large collaborations of scientists collect and analyze petabytes data from high-energy physics experiments, such as the Large Hadron Collider, hosted at the CERN laboratory in Geneva, Switzerland. Some of the questions that we collectively ask are:

- what are the fundamental blocks that make up our Universe?
- what is the nature of dark matter and dark energy?
- what is the nature of the asymmetry between matter and antimatter?
- what was the early Universe like?

To answer these questions, particle physicists build software to simulate and analyze what happens in particle physics detectors.

Since 2011, CERN has participated in other Google initiatives such as Google Summer of Code (GSoC), first as a small organization (CERN-SFT) and later as an umbrella organization (CERN-HSF) to involve the high-energy physics community.

Case Study Authors:

- David Lange (david.lange@princeton.edu)
- Vassil Vasilev (Vassil.Vassilev@cern.ch)
- Andy Buckley (andy.buckley@glasgow.ac.uk)
- Anders Kvellestad (anders.kvellestad@fys.uio.no)
- Christian Gutschow (chris.g@cern.ch)

Problem Statement

Particle physics relies heavily on open-source codes that compute key quantities for both theory and experiment. Often such codes have been developed for many years, with early documentation rendered incorrect or incomplete by later developments. Key user-facing information can be very difficult to find (e.g. in automatically generated code-interface documentation). It can be difficult for new physics-oriented users and contributors to get involved: the docs that exist are more for the developers and often assume too much technical prowess.

We seek to advance the documentation in two areas of high energy physics (HEP) research: interactive analysis based on C++; and reinterpretation of analysis results. Our intent is to build the sort of documentation that enables user engagement while being easy to update as our codes continue to evolve. In our community, the HEP Software Foundation (the “HSF”) has served as a hub for design discussions and project collaborations in these areas.

Proposal Abstract

Our aim was to advance the documentation in two areas of high energy physics (HEP) research: interactive analysis based on C++; and reinterpretation of analysis results.

Interactive Analysis: HEP researchers have developed several unique software technologies in the area of data analysis. Over the last decade we developed an interactive, interpretative C++ interpreter (aka REPL) as part of the ROOT data analysis project. We invested a significant effort to replace CINT, the C++ interpreter used until ROOT5, with a newly implemented REPL based on LLVM – Cling. Cling is a core component of ROOT and has been in production since 2014. Cling is also a standalone tool, which has a growing community outside of our field. It is recognized for enabling interactivity, dynamic interoperability and rapid prototyping capabilities for C++ developers. For example, if you are typing C++ in a Jupyter notebook you are using the xeus-cling Jupyter kernel. We are in the midst of an important project to address one of the major challenges to ensure Cling’s sustainability and to foster that growing community: moving most parts of Cling into LLVM. Since LLVM version 13 we have a version of Cling called Clang-Repl. As we advance the implementation and generalize its usage we aim for improving the overall documentation experience in the area of interactive C++.

Reinterpretation: “Reinterpretation” codes aim to reuse data to make new statements about fundamental physics. These codes are meant for wide use in the physics community, but for that to happen much better user-facing documentation needs to be created. We will focus on improvements to the Rivet and Gambit projects, two largely C++ physics applications for reinterpretation, which require a range of technical skills – from command-line Unix to C++ programming – in order to get the most from their capabilities.

Original proposal:
https://hepsoftwarefoundation.org/gsdocs/2022/proposal_analysis.html

Project Description

Creating the proposal

The HSF community solicited input from developers and project leads regarding places especially in need of improved documentation and with interested mentors to work with

GSoD participants. For example, groups discussed and identified the poor structure and outdated material on project websites as a barrier to uptake. In addition, opportunities for improving documentation to facilitate communication (both intra-team and to users) were identified.

Two topical areas were identified, as described above, out of this process. From these areas, a comprehensive proposal for a Google Season of Docs program was developed, socialized within the broader HSF community for comments and suggestions, and eventually submitted to the GSoD program after feedback had been included.

Budget

The budget and corresponding work scope was estimated based on prior experience and availability schedules of each mentor (eg, the reinterpretation work was constrained due to mentor unavailability once the fall semester started). The largest uncertainty was due to the regional variation in salary expectations of technical writers based on their location, which was uncertain at the time of the proposal. Nevertheless, the overall budget appeared to be appropriate given the time period and project scope that we envisioned.

Our budget turned out to be on target. Our expectations for technical writer productivity and costs roughly matched what we had planned for in our proposal. No additional funds outside of Season of Docs were used to fund the technical writers.

Participants

Interactive analysis component: Sara Bellei, Rohit Singh, Jun Zhang

Reinterpretation component: Anjelo Narendran (Rivet) and Ross Clark (GAMBIT, MCnet)

Mentors of each project component handled the recruitment and hiring of their technical writers separately. We describe each process and outcome separately.

Interactive Analysis component:Upon the selection of CERN-HSF as a GSoD participating entity, numerous potential technical writers contacted the mentoring team. We defined a process that included a small challenge exercise and followup discussion to allow the writers and mentors to get to know each other and to gauge interest and skill set. In the end, five candidates were interviewed and two selected. Sara Bellei who was a PhD scientist in Germany relatively new to technical writing; and Rohit Singh who was an undergraduate computer science student in India with a considerable background in technical documentation development. We intended that the two would complement each other as Rohit had much more expertise in technical aspects, and Sara more experience with scientific practices and needs. Unfortunately, Rohit was sick during the summer and then taken by courses in the fall, and made only a limited contribution before dropping out. However, Sara quickly gained the needed technical expertise to link her developments into the project websites and blogs. We are now considering one additional team member, Jun Zhang, an undergraduate student in China, who had worked with us over the summer and expressed interest in improving our technical documentation. This would occur after the current GSoD phase is complete.

All work happened remotely, using frequent team meetings and tools such as Slack to communicate when needed to solve technical roadblocks. It is clear that scheduling frequent interactions, including debugging sessions, was essential for getting new team members started and comfortable in asking questions and otherwise interacting with the full team.

Reinterpretation component*:* In the case of the reinterpretation component, both technical writers are physics students at University of Glasgow: it was not intended to select only local candidates, with the application process including advertising through the GAMBIT and MCnet collaborations' academic contacts as well as software-writers applying based on the project listing on the GSoD public pages. All applicants were asked to respond to three questions about the project motivation and how they planned to deal with some anticipated domain-specific issues and constraints.

The selection was made on judgment of how easily candidates would acquire the (limited, but not insignificant) physics context required to write useful tutorial material. Our experience from email exchanges and some video interviews was that most technical writers could not clearly answer how they would adapt to the physics context, and were fixated on a particular documentation system with which they had experience but which would not integrate with our code management: this felt like we would end up with some documentation not well suited to the audience, and which would rapidly decouple from the developing codebase. Several dropped out during the selection step.

The project work was largely supervised remotely, as both participants preferred to work from home much of the time, but we allocated office space in Glasgow for meetings with the supervisor (as well as virtually, especially for meetings with the wider international collaborations) and assisting each other.

Timeline

Our projects proceeded more or less on the schedule they had planned.

Interactive Analysis: Sara started in May and has continued until the end of the Seasons of Docs program. Rohit also started in May, but dropped out as explained above. Jun has only recently started and has agreed to continue past the end of the program given his interest in the work.

Reinterpretation: This component was planned to run from June-Sept 2022. One aspect of the project (GAMBIT documentation) started as planned; by mutual agreement the other part (Rivet) was delayed so the best candidate, who already had some experience with the platform, could complete a summer internship: his work in the remainder of the project was for a larger time-commitment so the work done was overall equivalent.

Results

Interactive analysis component: Work included organizing project blog posts, including those from summer student code contributors (via GSoC and other programs); organizing and consolidating existing materials; creating new documents based on existing tutorials and other technical information; and understanding how existing and new content should be organized across different open-source projects. The last aspect was particularly critical for us, as some project components span several open-source communities as it extends from a high-energy physics specific toolkit ("Cling") to one integrated into LLVM ("clang-repl").

Reinterpretation component: As a precursor to the GAMBIT documentation work, Clark was given the task of redeveloping the simpler MCnet (umbrella for Rivet and other open-source tools) website to explore the Hugo static-site engine's suitability for HEP software documentation. This structural update was successfully achieved, with content updates made by MCnet specialists to explore the maintenance model: the decision was to use Hugo also for the other projects, and to explore extensions to it for embedding code documentation.

Clark went on to develop an attractive and modern replacement website for GAMBIT, including extensions to some aspects of the Hugo style engine and the Doxybook2 interface to the XML output of the Doxygen documentation parser. He added an installation guide, a basic usage tutorial, and a guide for site maintenance as a handover record.

Narendran started project work later, picking up on several by-then established tropes for using Hugo and integrating with gitlab CI. He developed CI pipelines which made use of the latest complete, successful build of the Rivet code to export data files used by Doxybook and Hugo to render not just C++ code documentation, but modified versions of existing but unpublished tutorial material, and publication-related metadata generated using the compiled Rivet framework and git repository. Several scripts in Rivet were modified to make this work, and will be merged into the main analysis branch. Documentation of Python APIs is partially complete, with a stopgap solution until a more technically demanding version can be implemented by other project members and integrated into the new website/docs system.

Code contributions:
https://reviews.llvm.org/D138698
https://github.com/root-project/root/pull/11715
https://cling.readthedocs.io/en/latest/
https://gohugo.io/
https://github.com/matusnovak/doxybook2
https://hepmcnet.gitlab.io/
https://gambitbsm.github.io/
https://heprivet.gitlab.io/

Metrics

The primary metric used in both components was around user satisfaction and developer community acceptance. All technical writers regularly presented updates on their results and planned next steps in collaboration meetings. This facilitated constructive feedback as the projects were developed. As a result, the results have gained rapturous approval from our teams when finalized. It is clear to our community that the excellent documentation enhancements are a direct result of GSoD contributions.

Analysis

The mentors of all contributions agree that the overall effect has been extremely positive. The interactive analysis component landed several new documentation components, including its first documentation within the LLVM distribution. The reinterpretation component has gained excellent redevelopments of the web documentation for the two analysis-preservation and reinterpretation codes.

The more domain-specific parts went necessarily slower, and flagged that expert tutorials need to be written by experts rather than newcomer proxies; instead we focused on installation guides and absolute-beginner tutorials, which address issues that are hard for established developers to notice.

Overall we consider the project to have been a good success. For the reinterpretation component, this is reflected in that all three new websites will be made live imminently, once newly purchased domains have been activated and DNS configured. The CI-driven, static-site approach has turned out to be very usable by academic software collaborations, with just the right balance of technicality and simplicity.

Summary

The overall experience has been positive, here we comment on some specifics of our experience. The recruitment process was at times frustrating due to the majority of applicants not engaging with the nature of the software involved and rather expressing interest in many opportunities. This is not different from Google Summer of Code, but does require some substantial effort to work through interested technical writers. It is essential for projects to work with applicants to see that they are able to adapt to the modes of our particular user & developer community, and constraints of the existing codebase. Failing to do that is likely to negatively impact the quality of technical documentation produced during the project itself.

The content development was a positive experience both for the documentation developers and for us as project managers: both developers understood the ethos of our academic community, and were happy to adapt their own concepts of how things should look, work and be described, in response to feedback from the projects' members. The end results are extremely valuable updates to project websites, sustainably integrated with the codebases: more content needs to be added, but the work has been an essential contribution to making these codes accessible to new users. All of our project leaders found frequent communication, including informal means such as chat, important for their project components.