



LIETUVOS MOKSLO TARYBOS FINANSUOTO IMPRESS-U PROGRAMOS PROJEKTO

**„IRIS-HEP: MOKSLO PROGRAMINĖS ĮRANGOS PLĖTIMAS DALELIŲ FIZIKOS REIKMĖMS
UKRAINAI, LENKIJAI IR LIETUVA“**

(Nr. S-IMPRESSU-24-4, reg. data 2024-06-26 d.)

2025 M. STUDENTŲ STAŽUOČIŲ PATVIRTINTI PROJEKTAI

Projekto vykdytojas: Vilniaus Universitetas

Projekto vadovas: dr. Andrius Juodagalvis

2026 Vilnius

Adapting the Coffea Framework for FCC

Applicant: Eglė Lenkaitytė

Mentor: David Lange

Project duration: June 30th 2025 – September 2025 (12 weeks)

Project description

The Future Circular Collider (FCC) is a planned next-generation particle accelerator at CERN, supposed to be a successor to the Large Hadron Collider (LHC). With a potential to reach 100 TeV collision energies, increased data rates and more precision, FCC will open the way to greater understanding of fundamentals and the search for new physics beyond the Standard Model. Even though the FCC is not to start running for the next 15-20 years, it is crucial to develop robust data analysis infrastructure now to ensure good preparation for when the real data begins to flow [1].

The Columnar Object Framework For Effective Analysis (COFFEA) is a Python-based unified analysis toolkit for High Energy Physics (HEP) data. Unlike traditional event loop based HEP analysis, columnar analysis operates on data in a column-oriented, array-programming style. Visualisation of both is shown in Figure 1. Coffea provides tools and components that simplify this method, enabling streamlined and reproducible HEP workflows. This approach offers more efficiency, scalability and is faster and easier to use [2].

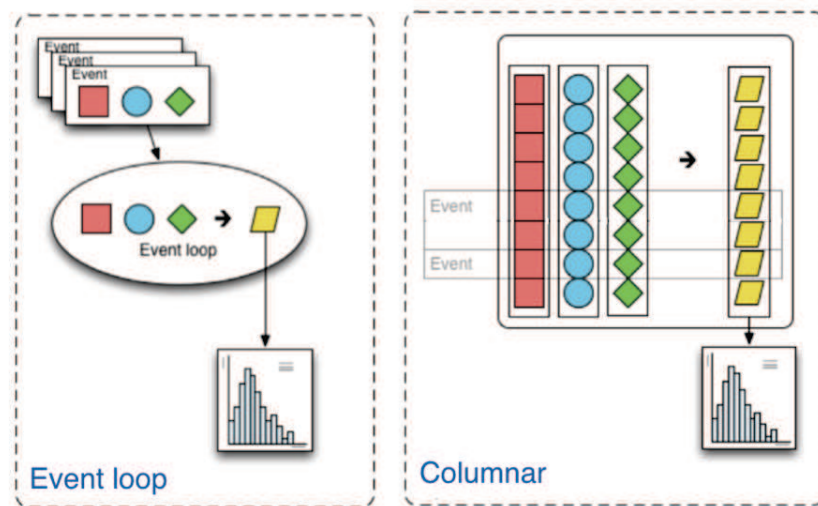


Figure 1. Schematic of the event loop and columnar data processing paradigm [2].

Coffea has already been successfully deployed in CMS (Compact Muon Solenoid). However, its application to FCC data, which is based on Key4hep [3] and FCC-specific software ecosystem (FCCSW) [4], is still being developed. This will allow for Coffea to work smoothly and be compatible with data formats and workflows used in FCC environment.

Planned work

This project aims to adapt the Coffea framework for use with FCC simulation data. Firstly, I will become familiar with the Coffea analysis framework. Secondly, I will create some examples of analysis using Coffea, initially basing them on existing materials, and later try including new examples that I develop. Thirdly, I will develop new types of algorithm components – such as particle identification or jet finding strategies – that are not yet implemented in Coffea.

Timeline

1-2 weeks: get familiar with Coffea.

3-4 weeks: build a series of example analysis for existing materials.

5-7 weeks: include my own examples into the analysis.

8-10 weeks: develop new types of algorithm components.

11-12 weeks: apply the components to my own examples and write documentation.

References

- [1] CERN. Future Circular Collider. URL: <https://home.cern/science/accelerators/future-circular-collider>
- [2] N. Smith, et al. Coffea – Columnar Object Framework for Effective Analysis. URL: <https://doi.org/10.48550/arXiv.2008.12712>
- [3] Key4hep documentation. URL: https://key4hep.github.io/key4hep-doc/getting_started/introduction.html#about
- [4] FCC Software overview. URL: <https://hep-fcc.github.io/FCCSW/>

Applicant: Rūta Lomsargytė
Mentor: Peter Elmer
Project Duration: 8 Weeks
Proposed Start Date: 1st July 2025

IRIS-HEP Fellows Program Project Proposal
Development of an LLM-Based Co-Pilot Tool for CMS
Analysts

Project description

In the recent years, Large Language Models (LLMs) have been helping users access complicated technical information and generate code, potentially saving time. In the context of CERN, CMS analysts often interact with specific software, internal documents and other computing tools. However, the internal documentation is not accessible to mainstream LLMs, therefore the information they output is not as accurate as it could be.

The aim of the project is to explore the possibilities of building an LLM-based co-pilot tool that assists CMS analysts. The tool would provide helpful answers to technical questions, assist in writing or understanding code, and retrieve relevant technical information from internal CMS sources. To reduce the need for fine-tuning and to optimize the use of computing resources, retrieval-augmented generation (RAG) technique will possibly be implemented. The project will focus on using internal CMS data while making sure that usage follows CERN policies.

Project goals

- Build an assistant tool that can:
 - Find and show relevant parts of CMS documentation.
 - Answer common questions about CMS analysis.
 - Help understand or write code snippets.
- Use CMS specific materials, including:
 - CMS Twikis, user guides, and other documents.
 - Code examples.
- Ensure safe and efficient usage by:
 - Using open-source or self-hosted large language models.

- Implementing retrieval-augmented generation (RAG) technique.
- Making sure the tool follows CERN data privacy policies.

Preliminary timeline

Weeks 1–2:

- Determine common tasks performed by CMS analysts that could benefit from LLM assistance.
- Set up the basic environment and create a plan for the rest of the project.

Weeks 3–5:

- Build a tool capable of understanding and responding to basic queries.
- Make the internal CMS documents accessible to the tool.
- Test the tool and correct based on feedback.

Weeks 6–8:

- Improve the tool’s ability to retrieve and output relevant information.
- Prepare a final demonstration and documentation of the tool.
- Draw conclusions for possible future improvements.

References

- [1] CERN openlab Technical Workshop 2025, “Development of Large Language Models at CERN,” Mar. 2025. [Online]. Available: https://indico.cern.ch/event/1440389/contributions/6364348/attachments/3024469/5339708/LLMs@CERN_OpenLabWS.pdf
- [2] CMS Collaboration, “CMS Software Documentation – The CMS Workbook,” [Online]. Available: <https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBook>

Backend update of the AICaVal framework for validation of calibration and alignment conditions

Fellow: Patrikas Styra

Mentors: Matheus Pereira Macedo De Sousa, Atanu Pathak, Andrea Perrotta

Duration: 8-12 weeks

Project Description

The alignment and calibration parameters of the CMS detector, also known as "conditions," are crucial for data quality and are stored in a database for online data acquisition and offline event reconstruction. These conditions continuously evolve due to factors such as faulty readout channels, detector aging, increased noise, and variations in sub-detector support systems (e.g., gas pressure, voltage, magnetic fields, and currents). Consequently, frequent updates and recalibration are necessary to ensure the recorded data accurately reflects real-life conditions.

Updated conditions must be validated before being appended to the database, along with their interval of validity. The AICaVal framework is vital for this validation task in the CMS experiment. While the current validation pipeline is functional, its speed can be improved, a need that will become more critical with the increased intensity of the LHC beams requiring a faster turnaround. Additionally, detailed performance monitoring is currently lacking.

This project will address these issues by:

1. **Optimizing the AICaVal backend pipeline:** The existing code for evaluating new conditions will be modified to reduce the overall processing time.
2. **Developing a monitoring dashboard:** Creating a visual tool to display statistics on how long each stage of the validation process takes and making it available to AICaVal users.

These enhancements aim to significantly improve the efficiency and monitoring of the AICaVal framework.

Software Deliverables

1. **Updated AICaVal Backend Code:** Modified scripts and classes for the AICaVal pipeline that show improved processing speed.
2. **Monitoring Dashboard:** A working web-based dashboard that displays timing information for the validation process statuses.
3. **Documentation:** Technical notes on the changes made to the pipeline and a guide for using the new dashboard.
4. **Final Report & Presentation:** A summary of the project work, results, and software produced, as required by IRIS-HEP.

Project Timeline (10 Weeks)

- **Weeks 1-3: Getting Started, Analysis & Early Development**
 - Learn the AICaVal system, focusing on the code of the validation pipeline.
 - Identify slow parts of the pipeline; begin making first improvements.
 - Plan the dashboard and how it will get data.

- **Weeks 4-7: Main Development Work**
 - Implement key changes to speed up the pipeline; measure the improvements.
 - Build the dashboard: data collection, processing, and visual display of timings.
- **Weeks 8-10: Testing, Writing Up & Finishing**
 - Test the faster pipeline and the new dashboard.
 - Write down the technical details and how to use the new tools.
 - Complete the final report and prepare the presentation.