DEVELOPMENTS IN FCCANALYSES

Status for June 2023

Juraj Smieško

CERN

FCC Software Meeting

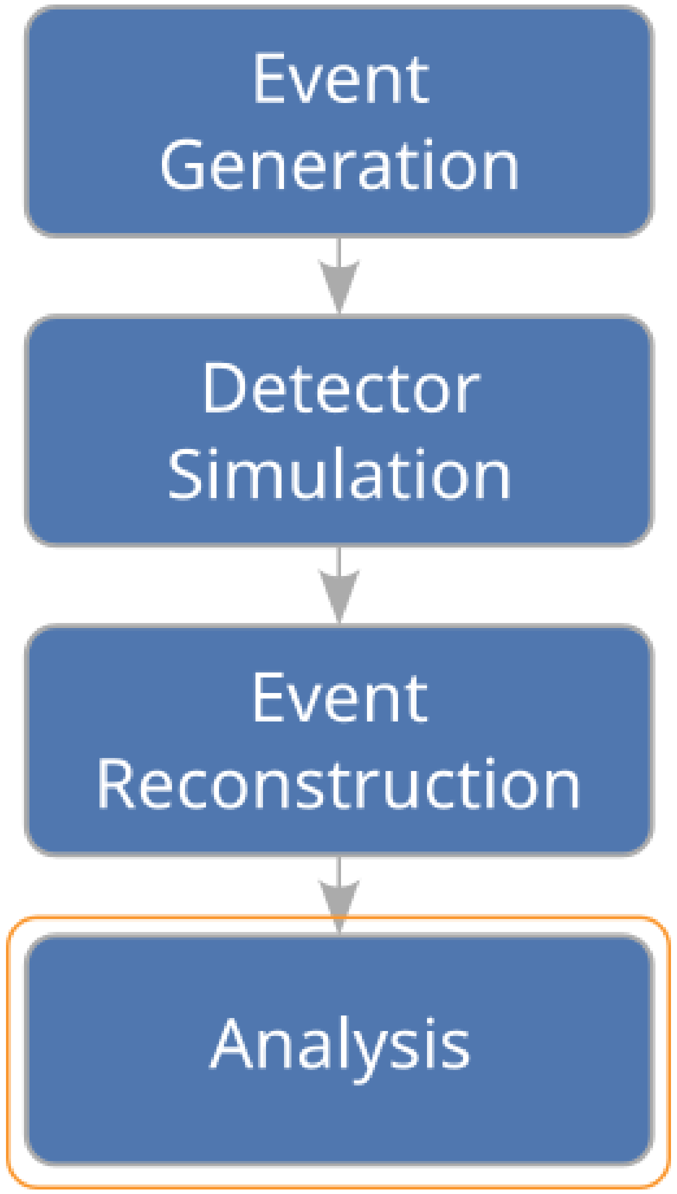
26 June 2023

FCCANALYSES SCOPE

Goal of the framework is to aid the user in obtaining the desired physics results from the reconstructed objects

Framework requirements:

- Efficiency — Make quick turn-around possible
- Flexibility — Allow heavy customization
- Ease of use — Should not be hard to start using
- Scalable — Handling of large datasets



KEY4HEP STATUS

Stable Key4hep stack:

- `/cvmfs/sw.hsf.org/spackages7/key4hep-stack/2023-04-08/x86_64-centos7-gcc11.2.0-opt/urwcv/setup.sh`
- Overview of Key4hep developments by Juan at EP R&D Software Working Group Meeting
  - EDM4hep Python bindings
  - podio: RNTuple backend on the way
  - Key4hep Validation
- Overview of Generators at Key4hep by Gerri at ECFA Higgs Factories: 2nd Topical Meeting on Generators
- Next FCC Physics Performance Meeting tomorrow

k4Geo

- FCCDetectors → k4geo migration has started!
  - PR#276 Noble Liquid ECAL barrel detector builder + documentation
    - Merged
  - PR#271 ARC (PID) barrel + endcap detector builders, compact file, examples, tests, documentation
    - Under review (well advanced, will likely be merged very soon)
  - PR#273 Detailed version of the vertex detector builder, embryo for full IDEA detector compact file, examples
    - Under review
  - Full Noble Liquid based concept compact migration starting
- To be addressed
  - Dual readout crystal ECAL
  - Dual readout fiber HCAL
  - Drift Chamber

FCCANALYSES STATUS

Merged PRs:

- FCCAnalyses #292: Fixed handling of empty outputDir
- Podio #412: Collection ID is now 32 bit hash
- Podio #427: Improved podio-dump

Issues:

- #291: Branch names for relations now easier to read

Upcoming

- k4FWCore #100: Podio Frame IO for Gaudi steering files
- WIP EDM4hepSource:
  Allows high level access to EDM4hep objects in ROOT RDataFrame

DATAMODEL EXPLORER

Simple JavaScript application to explore event MC tree: dmX

DOCUMENTATION & PLATFORMS

There are several sources of documentation

- FCC Tutorials: https://hep-fcc.github.io/fcc-tutorials/
  - Focused on providing a tutorial on a specific topic
- Code reference: https://hep-fcc.github.io/FCCAnalyses/doc/latest/index.html
  - Provides details about implementation of individual analyzers
- Manual pages:
  - Info about commands directly in the terminal: `man fccanalysis`
- FCCAnalyses website, FCCSW website

Please test Key4hep nightlies stack

- Three OSes supported: CentOS 7, AlmaLinux 9 and Ubuntu 22.04

BACKUP

FCCANALYSES VS. COFFEA/COFFEA-CASA

- Provides similar set of features to FCCAnalyses
- Dataframe in coffea, Orchestration in coffea-casa
- User interface purely pythonic
- Integrated into python package ecosystem
- FCCAnalysis purpose build for FCC
- Integration with SWAN and Dask

FCCANALYSES BATCH SUBMISSIONS

- FCCAnalyses allows users to submit their jobs onto HTCondor
- It bootstraps itself with use of scripts in subprocesses
- Framework creates two files
  - Shell script with `fccanalysis` command
  - Condor configuration file
- There is also possibility to add user provided Condor parameters
- Condor environment now isolated from machine where the submission was done
- Revised tracking across chunks/stages done with the variable in the ROOT file

SUB-COMMAND ROUTING

- There are three ways to run the analysis
  - `fccanalysis run my_analysis.py`
  - `python config/FCCAnalysesRun.py my_analysis.py`
    - Can this way be dropped?
  - `python my_analysis.py`
- Removed reliance on try/catch for sub-command routing

CODE FORMATTING

- Currently, there is wide range of styles used
- End goal: Make the analyzers better organized
  - They are building blocks of the analysis
- Created CI to check every commit
- LLVM Style selected based on popularity
- Only changed lines are checked

UPDATED VERTEXING

- Vertexing done with the help of code from Franco B.
- Introduces dependency on Delphes
- Introduces new analyzers: `SmearedTracksdNdx`, `SmearedTracksTOF`
- Simplifies Delphes–EDM4hep unit gymnastic
- Adds examples for B_S to D_S K

BUILDING OF FCCANALYSES

- FCCAnalyses is a package in the Key4hep stack
- Advanced users can work directly on their forks
  - Allows to keep the analysis ''cutting edge''
  - Requires discipline
- Added helper sub-command: `fccanalysis build`
- Current distribution mechanisms:
  - Using released version in Key4hep stack
  - Separate git repository + stable Key4hep stack
  - Separate git repository + nightlies stack

KEY4HEP STACK PIN

- FCCAnalyses is developed on top of Key4hep stack
- Sometimes depends on specific version of the package
- Added helper sub-command: `fccanalysis pin`
- Will pin the analysis to a specific version of the Key4hep stack
  - There is no patch mechanism in the Key4hep stack