Status and Plans for the FCC Software

2nd FCC-France workshop

Jan 20, 2021

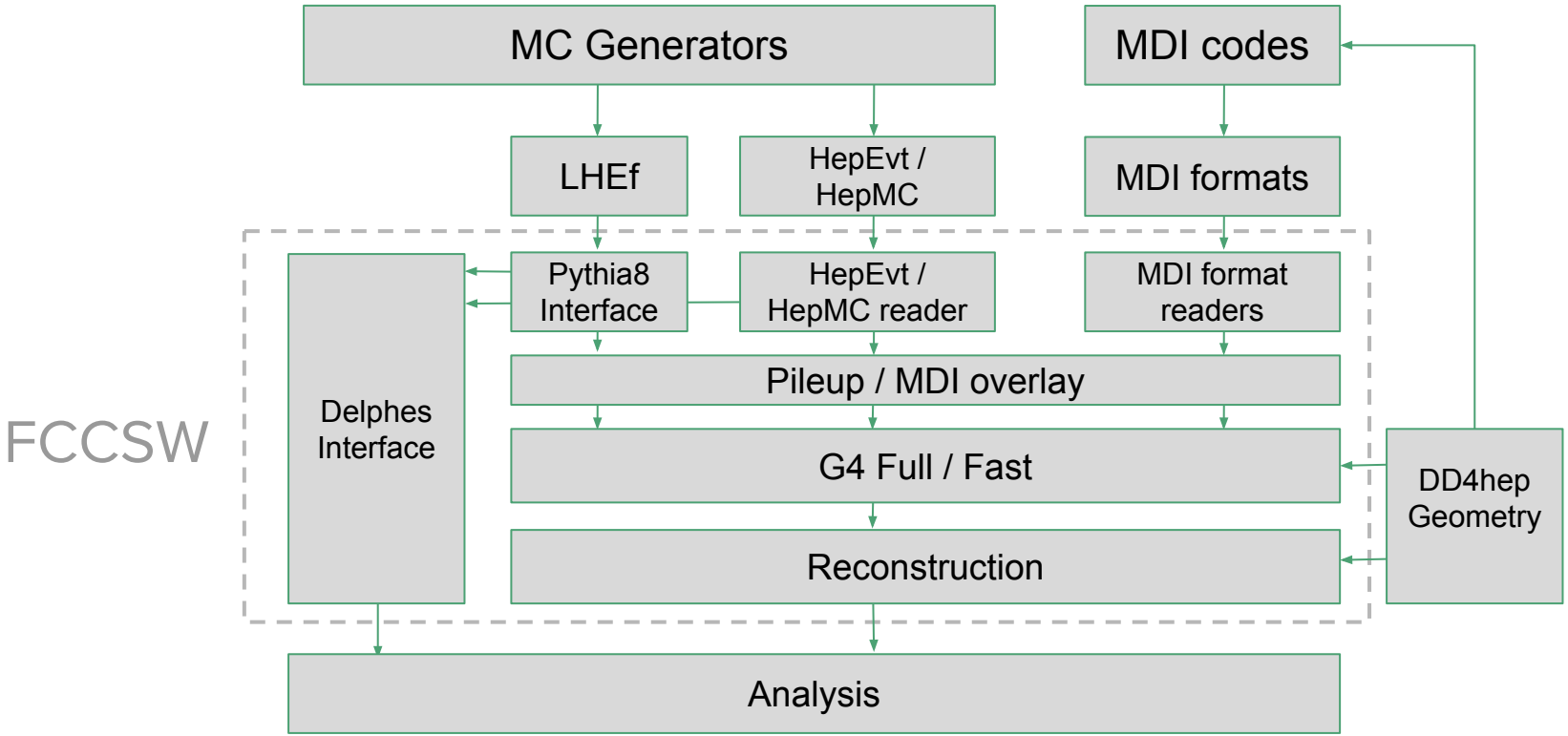
G Ganis, C Helsens
CERN-EP

FCC Software (FCCSW) reminder

Set of software tools to serve the needs of the FCC community

- Modular structure based on the Gaudi framework (LHCb, ATLAS)
  - Base of - and evolving in connection with - the Key4hep project
- Provides support for all the required functionality
  - Event Data Model (EDM), Generators, Geometry, Fast/Full simulation, Reconstruction, ...
- Served well the CDR. For the CDR++ required to improve
  - Available generators, in particular for FCC-ee
  - Palette of detector concepts with parametrized description ➜ Quality of the description
  - Palette of detectors with detailed geometry description ➜ Digitisation of their signal
  - Palette of reconstruction algorithms ➜ Integrate state of the art
- See presentation at last FCC France workshop

Typical workflows to support



What happened since last workshop - 1?

- EDM4hep consolidation
  - Complete set of data structures sufficient to replace FCC-edm
- Commissioning of a complete Delphes-based workflow
  - Significant improvements in Delphes, in particular the new TrackCovariance module
  - Enables advanced tracking (e.g. can develop vertexing algorithms) and the development of portable analysis techniques with EDM4hep
- Revisiting components Geant4 based full/fast simulation + Reconstruction
  - Augmenting detector palette (CLD fully available, LAr/tile calorimeter for FCC-ee)
  - Progress in integrating IDEA DR Calorimeter description to DD4hep
  - Enabled several good contacts with HGCAL, ACTS, SiPM digitisation, IDEA Drift chamber, etc...

What happened since last workshop - 2?

- Completion of Monte Carlo generators
  - KKMCee, Whizard, MadGraph5, BHLUMI (in progress)

- FCCAnalyses
  - Set of tools to help processing the output of ‘simulation’ in EDM4hep format
  - Close collaboration with the Physics Performance group

- Migration of the FCCSW framework to Key4hep / EDM4hep in progress
  - Isolate components of general interest (k4Sim, k4RecCalo, ...) to be moved to key4hep

- Tutorials / Starter Kit consolidation:
  - ReadTheDocs

Delphes update

Delphes: Parametrization of a Detector Concept

A framework for fast simulation of a generic collider experiment

C++ framework providing a fast multipurpose detector response simulation

- Tracking system, calorimeters and a muon system
- Includes:
  - Effects of magnetic field, granularity of calorimeters, sub-detector resolutions
  - Also observables such as isolated leptons, missing energy and collection of jets

Interfaced to standard file formats (e.g. Les Houches Event File or HepMC)

Detector concept palette includes FCC-hh baseline, IDEA, CLICdet, ...

Delphes in key4hep, output in EDM4hep: k4SimDelphes

- Set of standalone applications to cover more typical cases
  - {DelphesHepMC, DelphesSTDHEP, DelphesPythia8, DelphesROOT}_EDM4HEP
  - Pythia8 with EvtGen decays: DelphesPythia8EvtGen_EDM4HEP
- Gaudi framework integration in progress

FCCAnalyses and Delphes

FCCAnalyses - 1

https://github.com/HEP-FCC/FCCAnalyses/

Common tool for analyzing large datasets using RDataFrame and produce flat ntuples

It is composed of a library of C++ analysers and python configurations files

- C++ analysers are developed in common
- Python code specific to the analysis to define the analysers, output variables, input samples

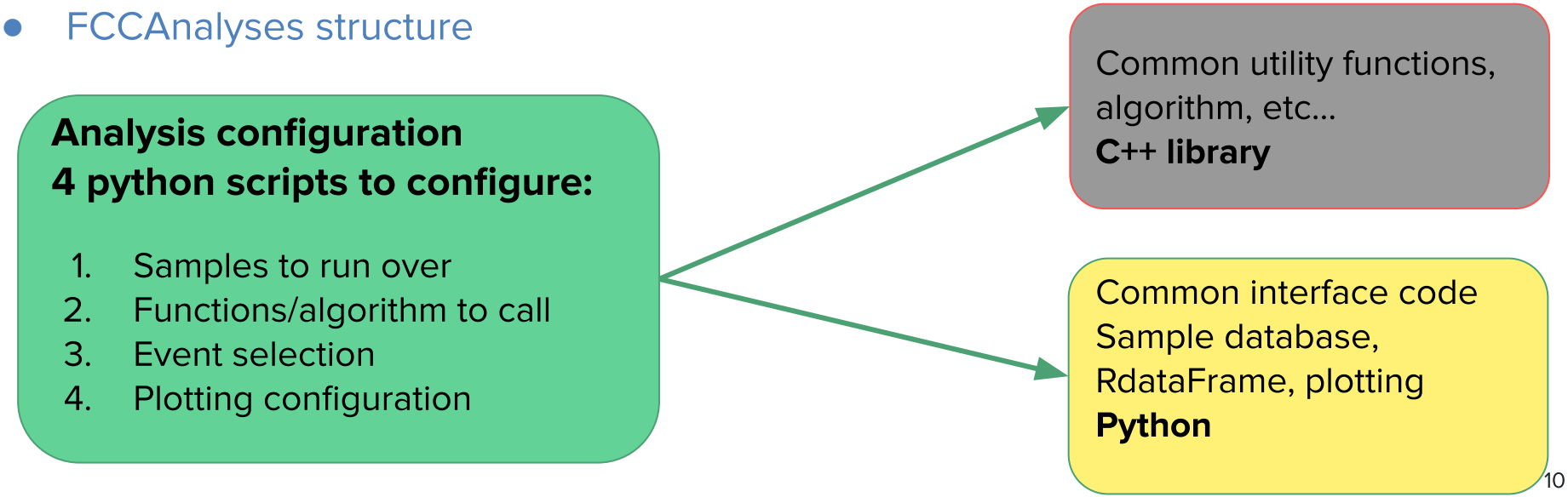
Flat ntuples are then used for example to:

- Produce variables for MVA training
- Produce final variables for analysis and plotting
- Run decay selector for flavour physics
- Etc...

FCCAnalyses - 2

https://github.com/HEP-FCC/FCCAnalyses/

- Set of tools to help processing the output of ‘simulation’
  - Agnostic to the type of simulation but specific reader functions are required
  - Build a common set of utility functions, algorithms for common use
  - Still possible for users to test their algorithms locally before publishing them
- FCCAnalyses structure



Workflow showcase: Flavour physics

- Use of EvtGen important in heavy flavour studies
- Use Pythia to generate $e^+e^- \rightarrow Z^0 \rightarrow bb$ and hadronise (make B-hadrons)

- EvtGen used to decay the hadrons produced
  - DECAY.DEC used in general to decay all of the products, but
  - User can request a specific exclusive decay chain for a particle e.g. $B^{\mp} \rightarrow (D^0 \rightarrow K^{\mp}\pi^{\pm})\pi^{\mp}$

- EvtGen is integrated in k4SimDelphes

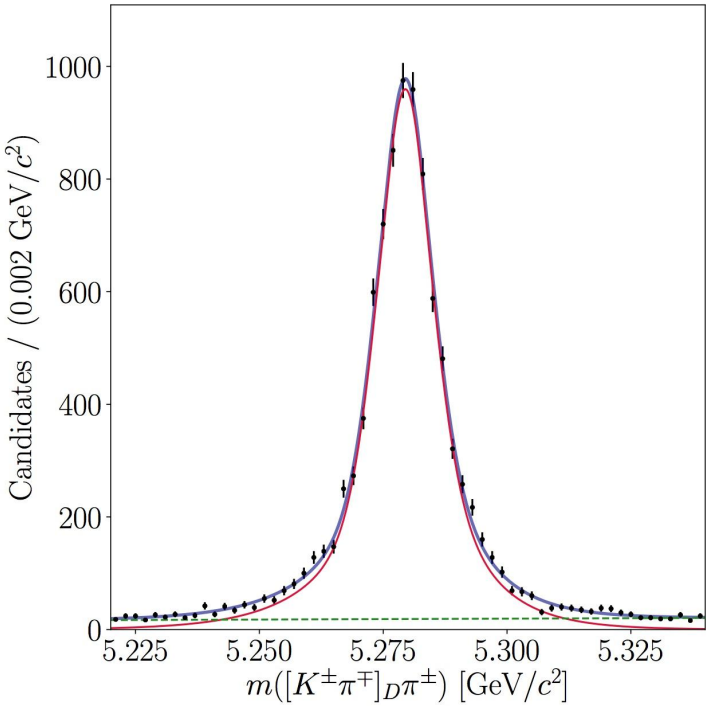
```
$ DelphesPythia8EvtGen_EDM4HEP  delphes.tcl  edmout.tcl ee_Z_bbbar.cmd out.root DECAY.DEC evt.pdl user.dec
```

- And also in FCCSW

```
$ fccrun PythiaDelphes_config_IDEAtrkCov.py -n 10000 --Filename ee_Z_bbbar.cmd --doEvtGenDecays true  \
    --EvtGenDecayFile DECAY.DEC --EvtGenParticleDataFile evt.pdl --UserDecayFile user.dec
```

Workflow showcase: Flavour physics

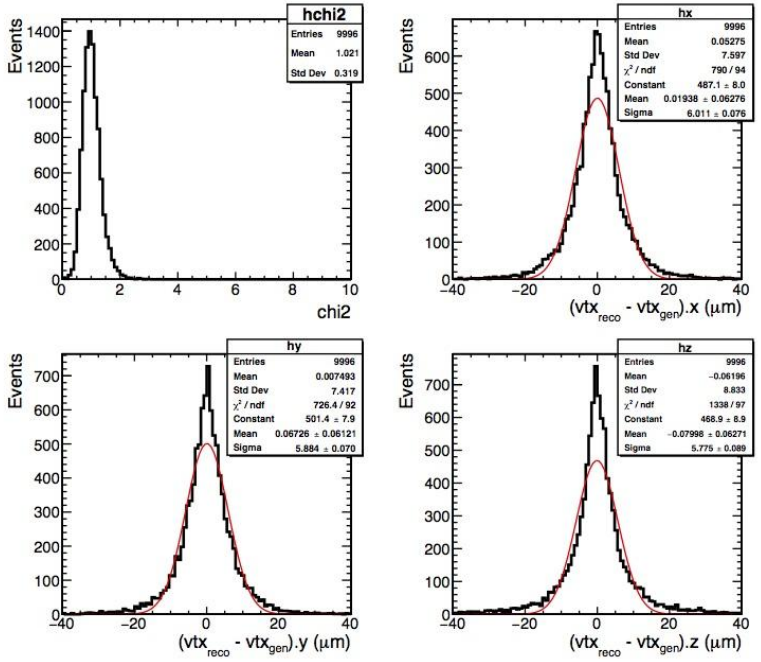
- 10k Z → bb with exclusive Bu2D0Pi.dec
  - 20k b-quarks
- 43% of q-quarks hadronise to $B^{\pm}$
  - expect ~ 8, 600 $B^{\pm}$ in total
- Using FCCAnalyses to produce small ntuples
- Using ReconstructedParticles to build the D^0 and $B^{\pm}$ cand.
  - Some loss expected due to track cuts in Delphes
- Fit $m(D^0\pi)$ in exclusive sample
  - Exponential background included in the fit
  - Only additional cut: ±30 MeV $m(D^0)$ window
- Yield well aligned with expectation



Workflow showcase: Vertexing

Vertexing using FCCAnalyses

- Vertices are not reconstructed in Delphes
  - Medium term plan: integrate LCFIPlus and ACTS
- Using Delphes events with track covariance
- Run a vertex fitting code written by F. Bedeschi and integrated in FCCAnalyses
- Gives the user the possibility to run vertex fitting on a subset of tracks, very useful for exclusive decays where you want to only use tracks from a reconstructed J/Psi and an other charged particle



Challenges ahead

Software and Computing

- Framework transition to Key4hep
  - Completeness of EDM4hep: stressing the EDM at the moment with use cases
  - Availability of relevant components: k4Sim, k4Reco, …
  - Algorithm availability and optimization
- Identify performance bottlenecks
  - Monte Carlo generators, in particular in corner cases (rare decays)
  - Full simulation components (DR calo …)
- Adequate and sustainable computing and storage infrastructure
  - Accessible and efficient meta-data, aggregation / sharing of simulated data files
  - Smooth access to distributed resources
  - Dataset reduction for efficient end user analysis

MDI-related software challenges

- Enabling estimation of reliable Machine-Detector effects
  - Identify needs in terms of multiturn, multiparticle tracking accelerator codes in particular with respect to
    - Beamstrahlung and beam-beam interactions
    - Machine imperfections and beam induced backgrounds relevant for the experiments

- Efficient and flexible interface of accelerator and experiments codes
  - Use of shared data formats

Dedicated working group being setup on this

Plans

Plans for 2021

- Core Software
  - Full migration to common software (key4hep) with comprehensive documentation
- Relation w/ Physics and Detector groups
  - Full understanding of Monte Carlo generators and decay tools
  - Geometry description for required sub-detectors
- Infrastructure
  - Prototype for ‘distributed’ resources (Dirac): file catalogue, access to computing
  - Estimation and procurement plan of needed resources
- Relation with MDI
  - Detailed plan and first prototype of a user-friendly meta-code for realistic estimation of machine-related backgrounds in FCCee

Overview of the areas of work

MC generators
Interfacing, testing, validating, optimization

Detector concepts
Geometry description, full simulation, validation, parametrization, optimization

Reconstruction algorithms
Tracking, vertexing, clustering, jet finding, particle identification, optimization

Analysis
State-of-Art (python) tools, ML, …

Computing
Porting to other OSs, Distributed computing, …

MDI
Shared formats, Identify relevant process and codes

Overview of the areas of work

MC generators
Interfacing, testing, validating, optimization

Detector concepts
Geometry description, full si
validation, parametrizat

igorithms
, clustering, jet finding,
ntification, optimization

Analysis
State-of-Art (python) tools, ML, …

ss,
nputing,
…

MDI
Shared formats, Identify relevant process and codes

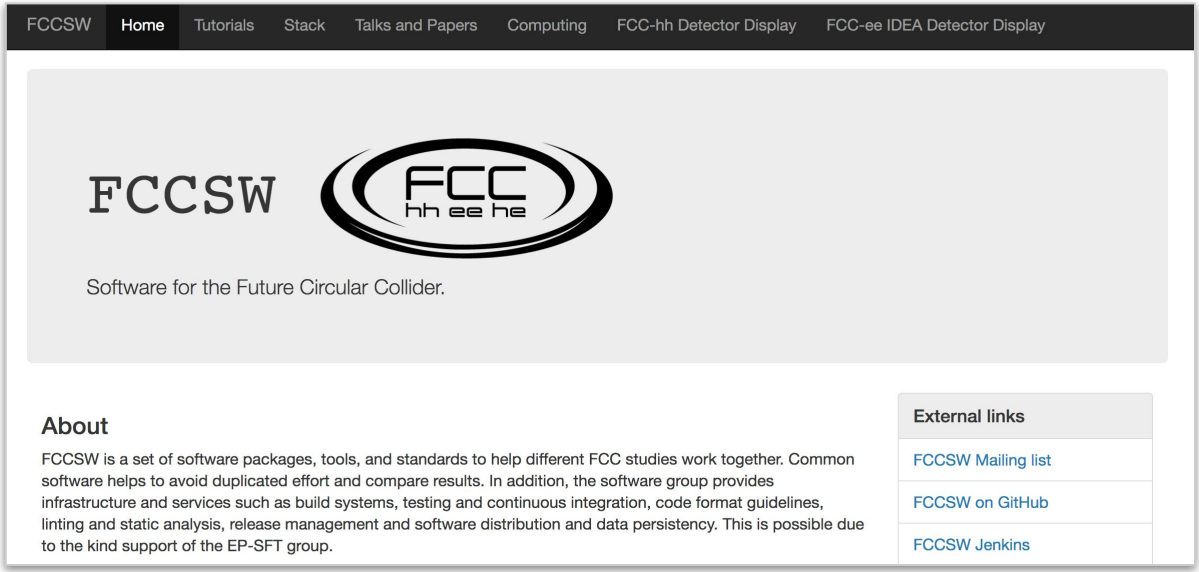
Nothing is over staffed, don't be shy please join!!

Summary

- Software is essential during any phase of a project
  - No CDR+/TDR without a robust software
  - Designing for long term usage provides stability and preserves knowledge
  - Essential role of documentation, training, for users and developers
- Current phase is challenging: working to make people able to contribute
  - Workflow based on Delphes fully available
  - Revisiting components Geant4 based full/fast simulation + Reconstruction
- Try to get as much as possible from the community
  - Following closely, participate-to, collaborate-w/ common activities {Key4hep, EDM4hep}
- Every group should feel concerned
  - Immediate areas of contribution identified
  - Output of European Strategy gave momentum, let's sustain it

Thank you!

- Web site https://cern.ch/fccsw



- Tutorials / Starter Kit: ReadTheDocs