A Software Framework for FCC studies: Status and Plans

CHEP 2019, Adelaide, Australia

Nov 04, 2019

N A Tehrani, J C Villanueva, J Faltova, G Ganis, C Helsens, J Hrdinka, C Neubüser, L Pezzotti, M Selvaggi, V Volkl, A Zaborwska
CERN
The FCC@CERN project

- Address main questions currently open in HEP
- Circular Collider of 97.75 km circumference
- **FCC-ee: \(e^+e^-\)**
  - \(\sqrt{s} = \{m_Z, 2m_W, 240, 2m_t\} \text{ GeV}\)
  - Data taking period: \(\approx 2040-2055\)
- **FCC-hh: \(pp, e^-p\)**
  - \(~100\text{ TeV} \text{ pp}\)
  - Possibly \(e^-p\) in parallel
  - Data taking period: \(\approx 2065-2090\)
FCC Conceptual Design Report

- Used as input for ESPP 2019
- Using FCCSW based on
  - 150 TB of full simulated data
  - 100 TB of Delphes events
- [https://cern.ch/fcc-cdr](https://cern.ch/fcc-cdr)
FCC Software Outlook

- Started in 2014, target the CDR
- Driving considerations
  - One software stack to support all the cases, all the detector concepts
    - Need to support broad range of event complexity ($e^+e^-, pp, e^-p$)
  - Need to support physics and detector studies
    - Parametrised, fast and full simulation (and mixture of the three)
  - Allow for evolution
    - Component parts can be improved separately
  - Allow multi-paradigm for analysis
    - C++ and Python at the same level
FCC Software Outlook

- **Adopted Strategy**
  - Adapt existing solutions from LHC
  - Look at ongoing common projects (AIDA)
  - Invest in streamlining of event data model

- **Future: towards a common software for future experiments**
  - [Bologna workshop, June 2019](#)
    - Present: LHC, ILC, CLIC, FCC, CEPC, SCTF, HSF
  - Agreed to:
    - Investigate the possibility to have a common event data model (EDM4hep)
    - Contribute to the development of a Common Turnkey Software Stack (Key4hep)
      - One framework (Gaudi best candidate), DD4hep, EDM4hep, Geant4, ROOT, ...
## FCC Software current ingredients

<table>
<thead>
<tr>
<th>Software Framework</th>
<th>Event Data Model</th>
<th>Detector Description</th>
<th>Generation / Simulation</th>
<th>Reconstruction</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>PODIO</td>
<td>DD4HEP</td>
<td>GenSer</td>
<td>G4</td>
<td>Delphes</td>
<td>HEPPY</td>
</tr>
<tr>
<td>Workload and Data Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Infrastructure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spack</td>
<td>LCG stacks</td>
<td>HEPPY</td>
<td>LCG stacks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CernVM-FS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HT-Condor</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EOS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Event Data Model and PODIO

- LHC experiments / LC studies solutions suffering (partly) from
  - Overly complex data models with deep object-hierarchies
  - Unfavorable I/O performance

- **PODIO**: an EDM toolkit (AIDA2020 project)
  - Plain Old Data, automatic code generation, support for different backends
    - Keep memory model simple, enabling fast I/O and efficient vectorization
    - Consistent / homogeneous implementation, minimizes mistakes
  - High-level description in YAML format
  - Three-layers
    - User: handles objects and collections
    - Object: transient, relations between objects
    - POD: actual data structures
  - Follow-up for next AIDA being prepared
    - Schema evolution, memory and I/O optimizations
Detector Description: DD4hep

- Generic detector view appropriate to support
  - simulation, reconstruction, analysis, ...

- Design goals
  - Complete
  - Single source of information
  - Easy of use

- Part of AIDA2020
- Used by CLIC, ILC, FCC, LHCb, CMS, SCT

F Gaede, T2 Tue, 16h30
DD4HEP and FCCSW

- FCC-hh
  - Reference detector complete

- FCC-ee
  - IDEA Concept
    - Beam Pipe, instrumentation
    - Vertex Detector, Drift Chamber
    - DREAM Calorimeter (under dev)
  - LAr+Tile calorimeter (under dev)
  - CLiCDP
Software Framework: Gaudi

● “The Gaudi project is a open project for providing the necessary interfaces and services for building HEP experiment frameworks in the domain of event data processing applications. The Gaudi framework is experiment independent.”

● Data processing framework designed to manage experiment workflows
  ○ Separate data and algorithms; well defined interfaces
  ○ User’s code encapsulated in Algorithm’s, Tool’s / Interface’s, Service’s
  ○ Different persistent and transient views of data
  ○ C++, with Python configuration

● Originating from LHCb, adopted by ATLAS, Daya Bay, GLAST, LZ, ...

● Actively developed to face LHC Run 3 and Run 4 challenges (high PU)
Gaudi and FCCSW

GENERATION
- Gaudi
  - Pythia...
- HepMC
  - Gaudi gen. config

SIMULATION
- Gaudi
  - Tracker Drift LAr-Cal
  - Gaudi sim. config
- DD4hep Geometry
  - Gaudi sim. config
- EDM
  - Gaudi reco. config

RECONSTRUCTION
- Gaudi
  - Tracker Drift LAr-Cal
  - EDM

FCC Software, CHEP2019
Monte Carlo Generation

- Interoperability with (Monte Carlo) generators at
  - Level 0 - common data formats
    - E.g. HepMC, Les Houches Event Files, MDI
  - Level 1 - callable interfaces
    - E.g. Pythia8
- Pythia8 also acts as LHEF event reader
  - Other converters available (HepMC, HepStd)
- Generator repository: GenSer @ LCG Stacks
  - Available in FCCSW
- MDI codes (MDISim, GuineaPig, ...) use level 0 interoperability
  - Adhoc converters under discussion
Simulation

- **Geant4**
  - Gaudi components exists to create
    - User Actions
    - Regions
    - Sensitive detectors
    - Selective output options
  - Mixing fast and full G4 simulation possible
    - SimG4Full / SimG4Fast

- **Delphes**
  - Gaudi interface
    - EDM output
Fast / Full Simulation Interplay

Example:
Higgs self-coupling
@ FCC-hh
Reconstruction

- **Challenges:** algorithm detector concept independent
  - Fully flexibility, avoid duplication

- **Tracking**
  - Track seeding (TrickTrack)
  - Hough Transform for drift chambers
  - Under implementation / investigation
    - ACTS integration
    - Conformal tracking

- **Calorimeters**
  - Sliding window (rectangular/ellipse)
  - Topo-clustering
Analysis considerations

- **HEPPY: High Energy Physics with PYthon**
  - Modular python framework for the analysis of collision events
- **Developed and still used for CMS**
- **In FCC is used to**
  - Process EDM events, apply-preselection, produce a flat and light ROOT ntuple
  - Analyse the ROOT ntuple
    - Not the only code used for this purpose
- **Flexible but slow**
  - Plan to move to a C++-based analysis framework, e.g. RDataFrame
Software Infrastructure

- Typical HEP development workflow
- Deliverables
  - FCCSW
  - Externals: FCCSW specific dependencies
  - Based on LCG releases
- Deployment on dedicated CernVM-FS repositories
  - /cvmfs/fcc.cern.ch, /cvmfs/fcc-nightlies.cern.ch
- Builds (nightlies, releases) managed by Spack
  - Good feedback to HSF packaging WG
About the after-CDR for FCCSW

● Requirements:
  ○ Support more detailed studies, in particular for e+e-, focusing on
    ■ *Completeness*: state-of-Art generators, MDI, reconstruction / analysis algorithms, ...
    ■ *Flexible detector description*: easy switch / replace sub-detectors, change dimensions / layout

● **Current approach** seems **adequate** to fulfill the requirements

● Foster and support
  ○ Participation from FCC institutes worldwide
  ○ Activities such as Key4HEP, which formalize and extended the FCCSW approach
Connection with Key4hep

<table>
<thead>
<tr>
<th>FCCSW</th>
<th>Detector Description</th>
<th>Key4hep</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific detector eg FCC-hh</td>
<td>IDEA CLiCDP</td>
<td>Pythia8</td>
<td>RDF PyHEP</td>
</tr>
<tr>
<td></td>
<td>Sil. Track. TPC</td>
<td>MadGraph GuineaPig</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DREAM</td>
<td>G4 Full</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>G4 Fast Delphes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TrickTrack</td>
<td></td>
</tr>
<tr>
<td>Specific algorithms</td>
<td></td>
<td>ACTS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conf. Track.</td>
<td></td>
</tr>
<tr>
<td>Specific effects eg Pile-Up</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCCSW</td>
<td>DD4hep</td>
<td>Gaudi</td>
<td>ROOT</td>
</tr>
<tr>
<td></td>
<td>EDM4hep</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Workload and Data Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Key4hep Software Infrastructure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FCC Software Infrastructure</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary

- The FCC software stack assembled using as much as possible existing components
- Served well the purposes of the CDRs
- Started a new phase to further develop to support more detailed studies, in particular for e+e-
- Follow closely, participate and collaborate new common activities {Key4hep, EDM4hep}
Thank you!

- [https://cern.ch/fccsw](https://cern.ch/fccsw)
Backup
Gaudi and FCCSW

- Python scripts to configure tools and algorithms and define their processing order

```bash
$ fccrun
  ./Examples/options/geant_fullsim_fccee_pgun.py
  --energyMin=10 --energyMax=10 --particleName="mu-"
  --n=1000 --filename=fccee_idea_mu.root
```

- Each algorithm defines its parameters which can be overwritten on the command line
Calo Reconstruction Example

- Single 100 GeV e- reconstructed by sliding window

- Single 100 GeV pion in 8+10 layers of the E+HCal reconstructed by topo-cluster