Summary of Software Workshop

Towards a common Turnkey Software Stack

Frank Gaede, DESY
IAS HEP Conference, Hongkong UST, January 21, 2020
Mini-Workshop  
Requirements of Software for e+e- Colliders

- one day workshop
- with software experts from all four e+e- Collider Projects: CEPC, CLIC, FCC and ILC
- follow-up on the Future Collider Software Workshop held in June 2019 in Bologna
- were the first steps towards a common software stack for all four e+e- colliders were taken
Introduction

Outcome of the Bologna Meeting

- many software experts from future collider collaboration and working groups: ILC, CLIC, FCC, CEPC, SCTF,…

- at the end of the workshop there was the overall agreement that we should try and work towards a common turnkey software stack: **Key4hep**

- based on:
  - standard HEP and external libraries like ROOT, Geant4, HepMC, VecGeom, boost, Eigen,…
  - **DD4hep, PODIO, ACTS, PandoraPFA,**…
  - new common Event Data Model: **EDM4hep**
  - use the primarily the **Gaudi** framework

Note that this is a **quite remarkable** outcome and at the same time a rather **ambitious** goal
The Turnkey Software Stack

work on Key4hep has started - mostly at CERN:

- **Manpower quest**
  - AIDA++ software submissions included Key4hep as R&D line
  - EP software R&D working package got concrete
    - A fellow hired, started Jan 2020 (V Volkl)
  - Second CERN fellow hired on CREMLIN PLUS funds
    - starting in March 2020

- **Dissemination of the idea**
  - Collaboration meetings: FCC, CLIC, ...
  - Talk at CHEP 2019

A generic Software Stack in HEP:

this needs to be filled with life using the agreed upon components
The FCC Software Framework

- FCC already has setup a software framework with most of the core components envisaged for Key4hep
- yet missing are reconstruction tools:
  - tracking tools
  - PandoraPFA
  - HLR tools: flavour tagging, jet finding, PID,…
- -> all of this is already available in the LC Marlin world
- and EDM4hep is not yet used (available)

open question:
should generator and simulation be run in Gaudi or standalone ?
-> should be flexible and allow both approaches in Key4hep
The ILC software framework

- the ILC community has started with common software tools already in 2003 with the development of the common event data model LCIO
  - four different experiments with four different frameworks at the time
- this has developed into iLCSof which is now used by ILC, CLICdp, CEPC, partly FCC-ee, Calice, LCTPC, HPS,…
- it uses the Marlin framework and DD4hep for the geometry description (and Geant4 simulation)
- a complete suite of digitisation, tracking, PFA and HLR tools exist
- these have been battle proven in several large Monte Carlo campaigns

it is important for the ILC community to preserve all the existing functionality when moving to Key4hep
The CLIC software framework

- CLICdp uses also the iLCSoft framework
- mostly the same algorithms as used for ILC, tuned and optimised for the CLIC detector
- also some specific algorithms for CLIC, e.g.
  - ConformalTracking: pattern recognition algorithm based on **conformal mapping** and **cellular automatons**
    - rather geometry agnostic -> could easily be adapted to other (Si-tracking) detectors
  - developed a programme based on **adiabatic changes** to move from iLCSoft to Key4hep
    - based on **Gaudi-Marlin wrapper**

also CLICdp community needs to preserve all the existing functionality when moving to Key4hep
The CEPC software framework

- started out from using **iLCSoft** (and ILD detector)
  - used for CDR (2018)
- now planning to use Key4hep eventually
  - started with using **pLCIO** and porting algorithms from Marlin to Gaudi
  - developed **LCIODataSVC -> pLCIO** object in memory
- ported **SiliconTracking** from **MarlinTrk** into Gaudi
- also ported **DDPlanarDigi** to DDPlanarSrv

very useful alternative route for the transition from **iLCSoft** to **Key4hep**
need to make sure that we soon integrate the different transition solutions from FCC, CLIC and CEPC
CEPC Simulation

- started to port the detector simulation model from the (old) iLCSoft Mokka simulation tool to DD4hep
- run the Geant4 simulation within the Gaudi framework
- developed a number of GenTools for reading generator information, conversion to MCParticle, a particle gun, etc
- Note: most of this and more plugins for MCTruth linking etc. are already available in DD4hep/DDG4 and used by ILC/CLIC \( \leftrightarrow \) need to understand differences
- CEPC also working on parallelising the simulation with MT and MPI (for HPC resources)
  - interesting for Key4hep

Design of simulation framework

- Based on Gaudi and DD4hep.
  - Reuse part of interfaces defined in FCCSW.
- In the current prototype, “Tracker” is setup.
DD4hep
Detector Description for all of HEP

- DD4hep originally developed in context of ILC/CLIC
- provides complete solution for detector description, simulation, interface to reconstruction, conditions and alignment
- large palette of sub-detector drivers exist - generic and CLICdp, ILD specific
  - to be extended as needed, e.g.
  - drift chamber and dual readout calorimeter
- DDRec: interface to geometry as needed for reconstruction
  - surfaces and generic detector classes
  - to be extended in a generic way
EDM4hep
The common Event Data Model for HEP

- the LCIO EDM has been a great success for the common software framework iLCSoft of the linear collider community
- it provided the basis for collaboration for more than 15 years
- EDM4hep should be a successor with modernised EDM and better I/O performance
  - based on LCIO and the fcc-edm
  - implemented using PODIO
- PODIO: EDM tool based on storing PODs with ROOT I/O
  - already used in the FCCSW and by pLCIO
  - pLCIO: almost complete re-implementation of LCIO in PODIO
- started with Simulation model: MCParticle SimTrackerHit, SimCalorimeterHit - next: tracking classes

involve experts from all four colliders in defining the EDM try to be close to LCIO to preserve all the existing code base
A worldwide Software Collaboration?

The LHC perspective on Key4hep

- LHC started with idea of having a common software stack - in the end only few common tools remained
- new common solutions arose with time, e.g: DD4hep
- HL-LHC is facing tremendous computing challenges
- HSF created to provide a platform addressing these challenges in a collaborative way - first outcome:
  - have started iris-HEP fellows programme in the US to address common HEP computing issues
  - could contribute to future collider community

Key4hep not targeted at HL-LHC - but we should of course stay close to LHC developments, even though our requirements are quite different
Conclusion

Summary of discussion at the end of the workshop

- workshop showed **quite some progress** in all four collider communities since Bologna meeting
- also many **open questions identified**, e.g. how to exactly go about the transition from iLCSof/Marlin to the new Gaudi based world
  - have seen quite different approaches
  - need to be flexible for quite some time
- however: general **agreement to continue the collaboration** and work towards a common Key4hep framework
- started to put together a long list of To-Do items in order to make progress fast

- **some immediate To-Do items:**
  - **goal:** get a first version of Key4hep with core component and a data service based on EDM4hep
    - setup Hey4hep mailing list
    - call for regular phone meetings
    - setup **Git project Key4hep** and add new packages there, e.g.
      - FWCore component, space installation scripts, experiment specific configurations
    - organise a dedicated software workshop in a few months from now

---

overall a very useful software workshop - but the real work is now ahead of us ...