FCC Software Overview

B. Hegner, CERN

for the FCC Experiment Software Team

LHeC Workshop

25 June 2015

Objectives and Considerations

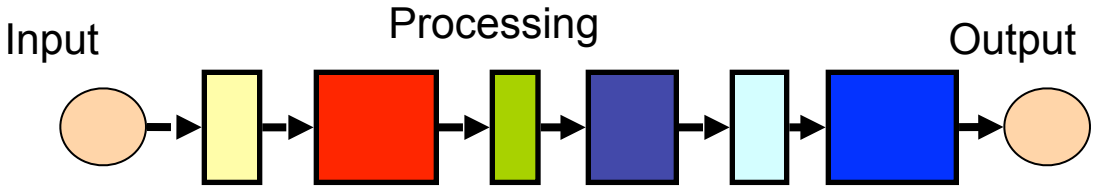
- Provide robust software to allow physics studies for CDR in 2018
- Support all **FCC-ee, -eh, and -hh** communities at the same time
  - Requires flexibility for Geometry and Simulation
- Start pragmatically
- As studies progress move to more sophisticated solutions
  - Allow components to be replaced later on
- FCC software effort relies on effort of other people
  - There is a give and take
  - Aim for, but don't blindly force, synergy with other communities

Early Decisions

- Adapt existing solutions from LHC
  - Gaudi as underlying framework
  - ROOT for I/O
  - Geant4 for simulation
  - C++ and Python for user analysis
- Adapt software developments from ILC/CLIC
  - DD4Hep for detector description
- Invest in **better fast vs. full sim integration**
  - Geant4 fastsim, Atlfast (ATLAS)
- Invest in **proper data model**
  - The LHC experiments' ones are over-engineered
  - The ILC/CLIC implementation (LCIO) isn't state of the art

Gaudi Event Processing Framework

- Gaudi is an event-independent data processing framework
  - Used by LHCb, ATLAS, and a few smaller experiments
- Based on the concept of a software bus
- Work is split up in interdependent "algorithms"



- All components developed will be plugged into here
- Parallelization effort with "GaudiHive" to take advantage of ever increasing hardware parallelization

Detector Description

Detector Description in LHC experiments is a not-well organized environment

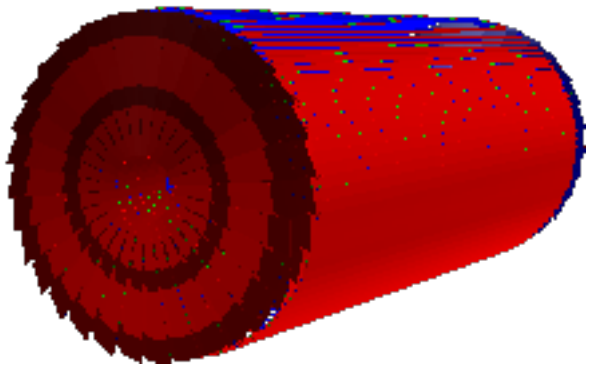
- Detectors modeled long ago and expertise largely gone
- Struggling themselves for the upgrade
- Heterogeneous setups even within experiments

ILC/CLIC efforts triggered the project DD4hep (*)

- Covering simulation, display, alignment in a consistent way

FCC joined these efforts of DD4hep

- Good support by developers!
- Good example of ILC-FCC cooperation
- **Addressing missing and duplicated functionality**



FCC Simulation

- FCC Software needs to support the studies of multiple detectors
- At different stages different level of detail required
  - Smearing vs. fast sim vs. full sim
- FCC choices are
  - Delphes (*)
  - Fast simulation
  - Full simulation with Geant4
- Should all be accessible from within the same framework

(*) http://delphes.hepforge.org

Delphes Status

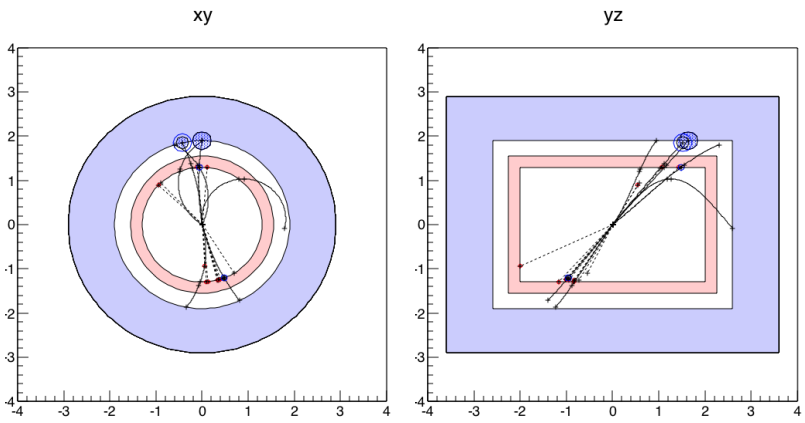
- Delphes has been (mostly) integrated into the FCC SW
  - Mapping of Delphes data types to FCC data types done
  - Expect a new Delphes version beginning of July for integration

    https://indico.cern.ch/event/399484/contribution/0/material/slides/0.pdf
- Preliminary Delphes data cards for FCC-hh prepared by Heather Gray and Filip Moortgat

- What about data cards for other use cases?

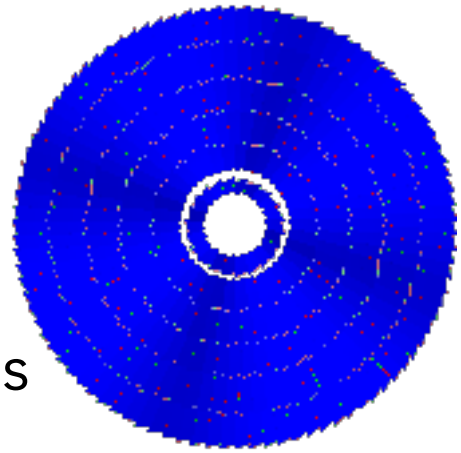
Fast Simulation - PAPAS

- PAPAS is a PArametrized PArticle Simulation package
  - based on particle flow experience mainly from CMS
  - prototyping environment for new algorithms in Python
  - 'integrated' into FCC software by using the same EDM
  - developed by Colin Bernet
- First test example yielded very promising results
  - under development to transform prototype into a more widely-applicable tool
- Focusses on FCC-ee though



Full and Fast Simulation

- Goal is to have a **combined fast and full simulation**
  - Decide at the config level where to do what
- (Semi-) automatic extraction of fast simulation parameters from full simulation
  - To be able to do fast-sim for any detector design
- Though not re-inventing the wheel, we are heavily re-designing it
- More details in one of the next presentations



Data Model I

The FCC requirements for a good data model are not special at all:

- Simplicity
- Flexibility
- Completeness
- Usable in C++ and Python

Data Models of LHC experiments are proven to work

- Fairly complex, and very detector specific beasts

The ILC community has a simple, but complete data model (LCIO)

- Needs adaption to allow direct ROOT access outside FWK
- Parallelism not part of the design
- Developers interested in extension and one should take advantage of it

The proper data model is **essential** **for allowing good results**

Thus we invested in a new project!

Data Model - the Library

- ROOT as first choice for I/O
- No deep object hierarchies
  - Wherever possible concrete types
- Simple memory layout
  - learning lessons from LHC
- Quick turnaround for improvements
  - Employ code generation
- Wrote a demonstrator data model
  - Used throughout all developments now
- No dependency of the data model on the experiment framework
  - easier support of use cases like FWLite, Python analysis or other standalone applications
- No assumptions about the physics data being stored

Data Model - the Library II

- Feature limited version of the library provided early this year
  - In use by multiple FCC software projects
- Second iteration of the library in progress
  - Prototype stage finished, but behind hoped schedule for putting it in production
- Discussions with LCIO developers whether this can be the next iteration LCIO implementation
  - Would mean share of manpower
  - Very encouraging so far

- The EU funded AIDA2020 program contains a data model deliverable
  - the FCC EDM library is now prime candidate for that effort
  - Distant future though

Data Model - the Data Types

A library is one thing - the definition of physics another!

Zillions of ~equivalent data definitions around.

- Contain almost the same physics content
- Are quite different in their organization

Spent some time to come up with a **tracking data model**

- Folding in experience from ATLAS, CMS and ILC
- Organizing data for easier access and usage given existing LHC code
- Track parameterization w/o assumption of a certain field
- **First iteration of tracking data model finished**

https://indico.cern.ch/event/400956/contribution/0/material/slides/0.pdf

- **More details on tracking later in this session**

Rest of the data model rather simple compared to it

- Preliminary definitions exist and are in use, but deserve a second iteration

Analysis

- Analysis should be easy and powerful
- Lesson from LHC experiments and ILC/CLIC
  - If data model too complex, physicists stop using common software and create their own mini-frameworks
- Need to allow multiple paradigms to do analysis
  - C++ and Python
- Physicists will join from different experiments and will bring along their existing code
  - **heppy** an example being brought into by CMS

Manpower situation

- Manpower still very critical
  - In particular lack of **expert knowledge** to take full advantage of the other volunteer work
  - E.g. no dedicated release manager to combine all the ongoing efforts into one single piece of software in a timely manner
  - Speed not limited by ideas, but by people contributing
- Manpower only slowly arriving:
  - one doctoral student for simulation (CERN FCC budget)
  - soon two more doctoral students
  - Senior fellow dedicated to SW only starting late fall

Next steps and related projects

- There is a huge list of items to tackle
  - Complete common EDM development and definition
  - Merge all simulation development streams back into Gaudi-driven framework (see rest of this session!)
  - Reconstruction development just starting
  - Creating a proper test suite
- Many software efforts going on in parallel
  - DD4hep under heavy development
  - Data model library part of AIDA2020 program
  - Gaudi-modernization effort between FCC/LHCb/ATLAS

Where are we now?

- Ideas are getting turned into real code
  - Fast/full sim design validated and being turned into real code
  - Data model library in 2nd iteration
  - Python analysis interface available
  - …
- Details being discussed (almost) every week Thursday noon:

  https://indico.cern.ch/category/5666/
- … and on our mailing list

  fcc-experiments-sw-dev@cern.ch
- **Please sign up and join!**