Gaudi and the FCC Software

Benedikt Hegner

Gaudi Developers Meeting
11.7.2015

The FCC SW effort

- 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

Backward compatibility

- Backward compatibility obviously a non-issue for us
  - We'd like clean interfaces w/o historical remnants
- We should drop
  - Historical duplications of services
  - Patterns invalidated by multithreading (e.g. various categories of tools)
  - Configuration options/syntax not matching GaudiHive (sequence unrolling is a hack!)
- We should update to more modern C++
  - Namespaces, more STL, smart pointers, ...
- We could take advantage of the [[deprecated]] attribute at many places and introduce a new deprecation warning for Components
- We understand that adiabatic changes are easier for running experiments...

  ... but we should make sure that we do not spoil the migration end-point by doing this

What is 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." - Gaudi website

- The most important question to us is - what is the aim of the Gaudi project?
  - currently it is for building an ***experiment framework***
- That is not enough for FCC!
- In our vision there should be the **building blocks for an entire HEP application** (turn-key system), including
  - Generation, Simulation, Detector Description
  - Interfaces to common physics libraries (e.g.fastjet)
  - Basic data models to communicate between components
  - ...
- The development will happen in any case, but where should the code go?
  - Extending Gaudi or creating a new project?

CPU technology

- **Following the GaudiHive approach through the entire code base…**
  … and making all services thread-safe

- **What's our vision concerning platforms, heterogeneity and multi-process communication?**

Organizing the Development

- How do we tackle backwards compatibility vs. making rapid progress?
  - Two main branches - legacy-branch and future-branch
    - More work for experiments
  - One main development branch
    - Slows down deprecation progress
- The internal repository organization doesn't follow an easy-to-understand structure
  - should dare moving things around
- Merging of feature pull requests by four-eyes/two collaborations principle
  - reducing the risk of writing too experiment specific code
- Major designs are decided upon in this very meeting (in particular all interface classes)
  - No new feature without JIRA ticket
- Time for a new design document as HSF technical note?

Our contribution

- We are willing to contribute with
  - fractions of my time
  - fraction of a senior fellow and a doctoral student (from end of 2015)
- Contributions to the very core of Gaudi
  - Multi-thread compatible services and tools
- Contributions w/ an extended feature set
  - Pick-and-choose of existing code or adding our own for a turn-key system
- Library services for the future branch
  - We need to do that anyhow for FCC
- We will not contribute to the legacy branch
  - we are not users of it and cannot judge the impact of changes