

Task 12.2 - Turnkey Software

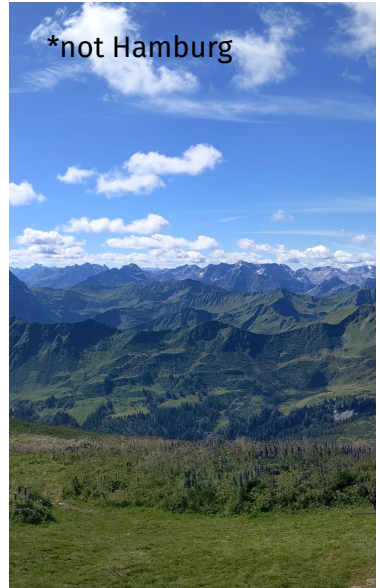
WP12 General Meeting

Thomas Madlener

Aug 26, 2021


General news

- Andre Sailer (CERN) is new co-task leader/deputy
- Holiday season: Nothing else to report from the administrative side



Latest developments in the Key4hep stack

k4SimDelphes

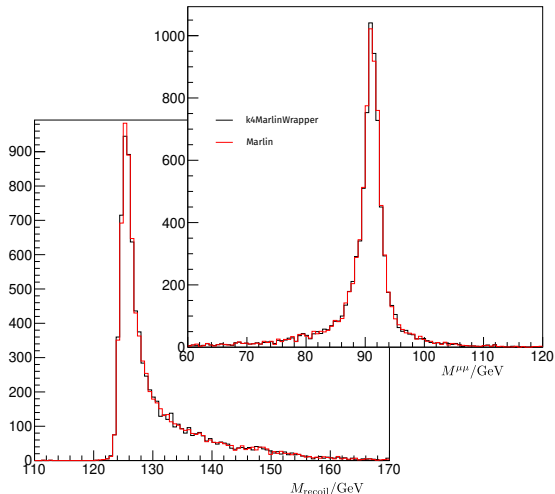
- Fixed some small issues in framework integration and added example running pythia and Delphes inside the Key4hep framework ( [key4hep/k4SimDelphes#63](https://github.com/key4hep/k4SimDelphes#63))

k4MarlinWrapper

- Identifying and fixing small issues
 - Differences between LCIO and EDM4hep data models and approaches
 - Handling of “EOF” and random seed
 - Some small differences in XML handling between Marlin and steering file converter script

ILD standard reco in k4MarlinWrapper

- Summerstudent project @DESY
- Compare results from standard Marlin and k4MarlinWrapper
- Producing LCIO output with k4MarlinWrapper is straight forward
 - Steering file conversion might need some “pre-processing”
- Some small differences between Marlin and k4MarlinWrapper even with fixed RandomSeed
 - Have to investigate the origins of these
- Still need to look at conversion to EDM4hep



LCIO vs EDM4hep - current issues

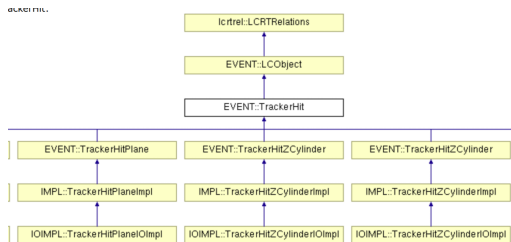
Relations

- LCIO allows for generic relations between any two collections
- EDM4hep introduces dedicated Association classes

```
edm4hep::MCRecoParticleAssociation:  
Description: "Used to keep track of the correspondence between MC and reconstructed particles"  
Author: "C. Bernet, B. Hegner"  
Members:  
- float weight // weight of this association  
OneToOneRelations :  
- edm4hep::ReconstructedParticle rec // reference to the reconstructed particle  
- edm4hep::MCParticle sim // reference to the Monte-Carlo particle
```

Inheritance and Value Semantics

- LCIO has an abstract TrackerHit interface
- EDM4hep only has concrete TrackerHit datatype
- Virtual inheritance not possible with value semantics
- Currently main issue in conversion of tracks
- Idea for generic wrapper types based on `std::variant`



Latest developments in podio & EDM4hep

- Implemented “subset” collections ( [AIDASoft/podio#197](https://github.com/AIDASoft/podio#197))

edm4hep::RecoParticleRef:

```
Description: "Used to get a subset of reconstructed particles from a collection (or many collections)"
Author: "T. Madlener, DESY"
Members: {}
OneToOneRelations:
  - edm4hep::ReconstructedParticle particle // reference to the reconstructed particle
```

```
// FILLING
auto& recos = store.create<edm4hep::ReconstructedParticleCollection>("recos");
// fill

auto& muons = store.create<edm4hep::ReconstructedParticleCollection>("muons");
muons.setSubsetCollection(); // declare this as a subset collection
// Here I can only store objects already tracked by another collection
muons.push_back(recos[0]);

// READING
auto& muons = store.get<edm4hep::ReconstructedParticleCollection>("muons");
for (auto muon : muons) { /* use as ConstReconstructedParticle */ }
```

- Previously needed a dedicated datatype in EDM4hep to store “references” to reco particles
 - Awkward to use in filling and reading
- Now much simpler to use
- Cleaner separation of storage concerns and collection interface (as a necessary side product)

Generic user data

- Not all data that needs to be stored fits into the EDM
- Need a way to store generic user data
- First idea: Define a generic datatype for the EDM that allows to store user data
 - For single values too much memory overhead
 - For multiple values very similar to GenericParameters of LCIO
 - “Last resort”
- Difficult balance between “easy to use” and “powerful enough to implement a parallel EDM”
- Current plan: See if podio can be made to handle `std::vectors` of fundamental types

```
edm4hep::UserFloat:
  Description: "A simple struct with one user defined float value"
  Author : "F.Gaede"
  Members:
    - float value // the single float value for extra user data

edm4hep::UserExt:
  Description: "A simple struct with user defined int/float/double"
  Author : "Tao Lin"
  VectorMembers:
    - int valI // data int
    - float valF // data float
    - double valD // data double
```

Summary

- Steady progress for Key4hep
- Challenges remain