Task 12.2 - Turnkey Software

AIDAinnova WP12 Meeting

Thomas Madlener

Jul 03, 2024

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 101004761.

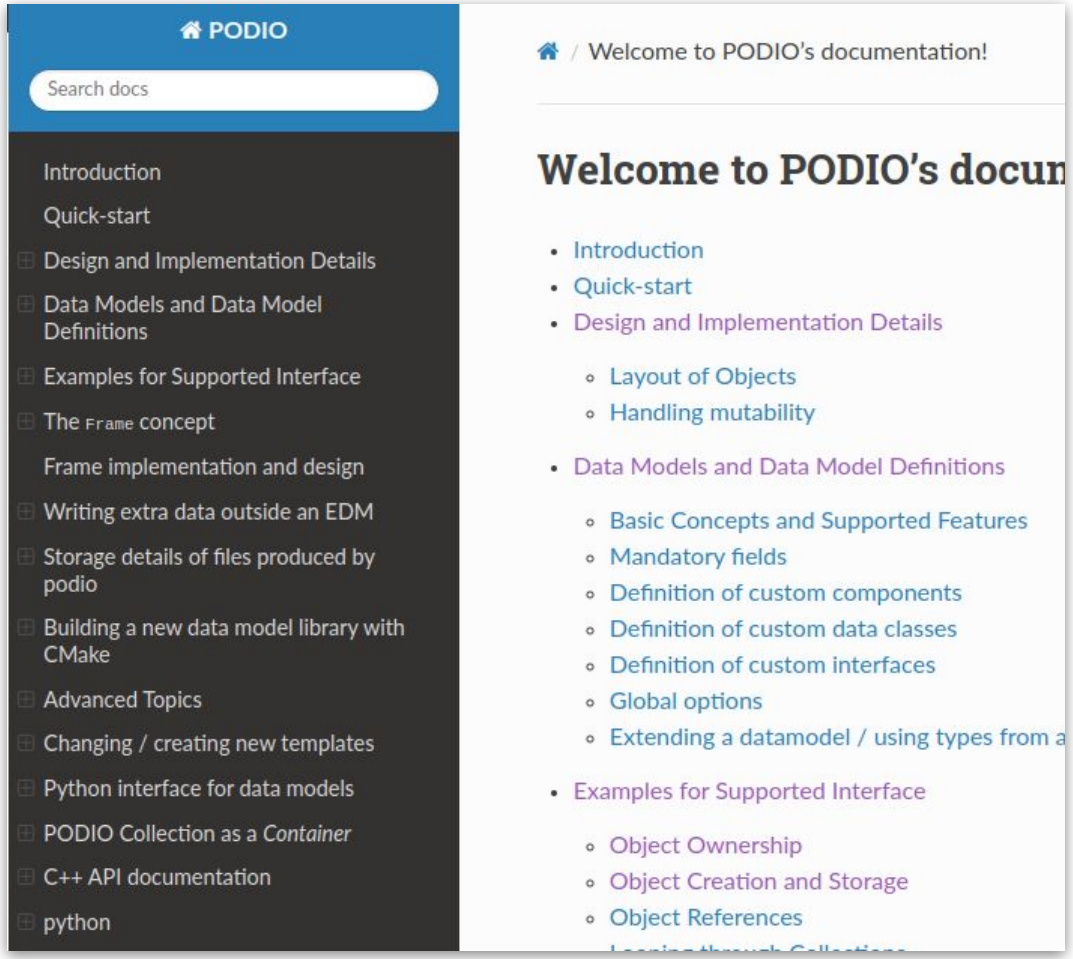
AIDA innova

CERN

DESY.

Developments in podio

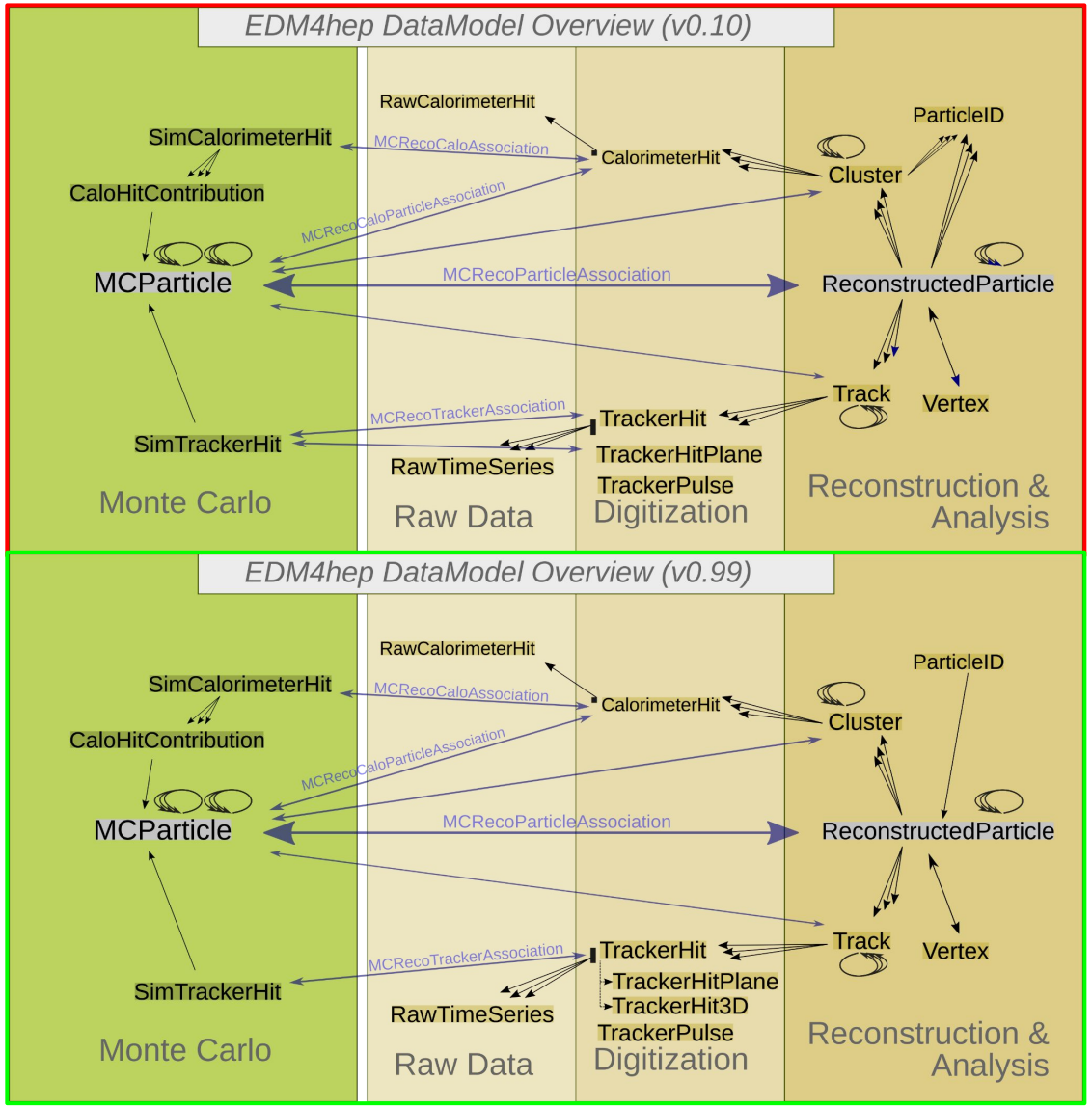
- **Version 1.0** (and 1.0.1) (see here for a more detailed list)
- Lots of documentation updates
  - Deployed to key4hep.web.cern.ch/podio
  - Compatibility between collection and *STL container* (#598)
- Few (small-ish) bug fixes
  - Schema evolution detects more impossible evolutions (#622)
  - Fix relations in cloned objects (#583)
- `Reader` and `Writer` "interface" (#522)
- Some API "fixes" (i.e. breaking changes)
  - `Frame::getParameter` returns `std::optional` (#580)
  - `clone` can now skip relations (#609)
  - Homogenize storage of parameters for TTree & RNTuple(#625)
- On the list for v1.1 (or beyond)
  - `RDataSource` for podio (#593)
  - Still a list of open issues and ideas :)



Developments in EDM4hep

Things that were done

- Many developments towards v1.0 - **almost done**
  - Quite a few breaking changes
  - Introducing conceptual differences wrt LCIO
  - Check EDM4hep v1.0 project for more details
  - Changes in edm4hep.yaml are trivial, but usually several repositories affected downstream
- New datatypes for generator info (#310)
- Reverse the direction of the ParticleID relation (#268)
  - Including utilities for resolving relations again
- Dedicated covariance matrix components (#287)
- Remove some of the drift chamber study types (#333)
  - Needs a bit more design work to be better integrated into the overall structure
- Remove TrackerHitPlane - SimTrackerHit assoc. (#331)
- Change probability to ndf in Vertex (#324)



Developments in EDM4hep

Things that are still ongoing

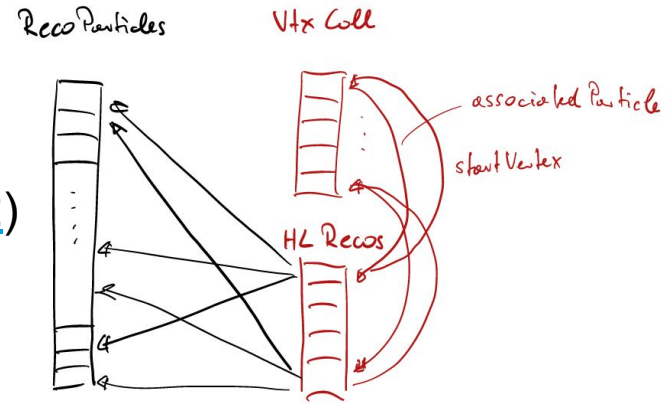
- Make `primary -> type` bitfield in Vertex (#329)
  - Consistent with what we do in other types
- Remove radiusOfInnerMostHit from Track (#326)
  - Part of an overall goal to not mix analysis and reconstruction level quantities too much
- Remove the dQ/dx information from Track (#311)
  - Not really usable in multithreaded context
- Refine Vertex - ReconstructedParticle relations (#332)
  - Should become a bit more intuitive to use
- Plan to tackle them this / next week
- Make pre-release tag before summer break
- Some documentation still to do
- v1.0 after summer (before CHEP)

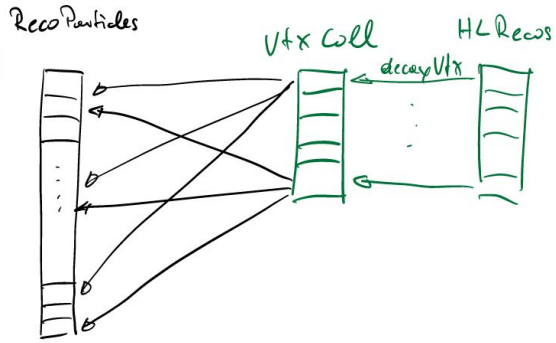
Vertex:

```
-     OneToOneRelations:
-       - edm4hep::ReconstructedParticle associatedParticle //
+     OneToManyRelations:
+       - edm4hep::ReconstructedParticle particles // particles
```

ReconstructedParticle:

```
      OneToOneRelations:
-       - edm4hep::Vertex          startVertex    // start vertex
+       - edm4hep::Vertex          decayVertex    // decay vertex
      OneToManyRelations:
```





Developments in k4FWCore

- Improved errors from k4run (#178)
- Make `MetaDataHandle::get` throw on missing values (#195)
- Support for multithreading (#173, #201)
  - Introduce dedicated `IOSvc` & `Reader` / `Writer` algorithms (to replace `Podio[In|Out]put`)
  - Custom implementations of `Consumer`, `Producer`, `Transformer` (of Gaudi) to integrate with podio
  - Support for variable number of input collections
- Tested for mixing & matching of “legacy” algorithms and functional ones
  - Including wrapped Marlin processors
- Required features coming from porting DDMarlinPandora processors to Gaudi

#201 (wip)

```
struct ExampleFunctionalConsumerRuntimeCollections final
    : k4FWCore::Consumer<void(const std::vector<const edm4hep::MCParticleCollection*>& input)> {
  // The pair in KeyValue can be changed from python and it corresponds
  // to the name of the output collection
  ExampleFunctionalConsumerRuntimeCollections(const std::string& name, ISvcLocator* svcLoc)
      : Consumer(name, svcLoc, KeyValues("InputCollection", {"DefaultValue"})) {}

  // This is the function that will be called to produce the data
  void operator()(const std::vector<const edm4hep::MCParticleCollection*>& input) const override {
    if (input.size() != 3) {
      throw std::runtime_error("Wrong size of the input vector, expected 3, got " + std::to_string(input.size()));
    }
    for (int i = 0; i < 3; i++) {
      if (inputLocations(0)[i] != "MCParticles" + std::to_string(i)) {
        throw std::runtime_error("Wrong name of the input collection, expected MCParticles" + std::to_string(i) +
                                 ", got " + std::string(inputLocations(0)[i]));
      }
    }
```

```
consumer = ExampleFunctionalConsumerRuntimeCollections(
    "Consumer",
    # InputCollection="MCParticles0,MCParticles1,MCParticles2",
    InputCollection=["MCParticles0", "MCParticles1", "MCParticles2"],
    Offset=0,
)
```

Key4hep stack news

- DDFastShowerML part of they key4hep stack (nightlies)
  - Result of hackathon at annual meeting
- New MarlinMLFlavorTagging package (nightlies)
  - Developed at DESY
  - First prototype of a somewhat generic inference interface
  - Can serve as example and basis for similar things in Gaudi(?)
- New k4Reco package (nightlies)
  - Home for generic reco algorithms (digitizer atm)
- New k4DetPerformance package (not yet part of stack)
  - Home for generic performance study script (currently tracking only)
  - Started from G. Sadowski's studies
- New k4GeneratorsConfig package (not yet part of stack)
  - Automatic generation of input files for generators from one source
  - Part of ECFA study
  - Uncovered several generator (build) issues already