

Podio Collections in Acts

Fitter, happier, more productive

Current Situation

edm4hep::SimTrackerHit -(SiliconTrackerDigi)--> edm4eic::RawTrackerHit + Ass

edm4hep::RawTrackerHit -(TrackerHitReconstruction)--> edm4eic::TrackerHit

edm4eic::TrackerHit -(TrackSeeding)--> edm4eic::TrackParameters

edm4eic::TrackerHit -(TrackerMeas'ntFromHits)--> edm4eic::Measurement2D

edm4eic::Measurement2D + edm4eic::TrackParameters -(CKFTracking)-->

edm4eic::Trajectory + edm4eic::TrackParameters + edm4eic::Track +

ActsExamples::Trajectories + ActsExamples::ConstTrackContainer

ActsExamples::Trajectories + ActsExamples::ConstTrackContainer

-(TrackPropagation)--> edm4eic::TrackSegment

Drawbacks of Current Situation

- Even if we had associations throughout, or a full chain of relations from Measurement2D to TrackerHit to RawTrackerHit (and an association to MCParticle), we still don't have this through CKFTracking
- ActsExamples::Trajectories and ActsExamples::ConstTrackContainer are not podio types, so cannot be linked to from associations and/or relations
- Inside CKFTracking, we have to copy
 - edm4eic::Measurement → ActsExamples::MeasurementContainer + ActsExamples::IndexSourceLink
 - edm4eic::TrackParameters → ActsExamples::TrackParametersContainer
 - ActsExamples::Trajectories → edm4eic::Trajectory
 - ActsExamples::ConstTrackContainer → edm4eic::Tracks

What is ActsExamples::ConstTrackContainer?

```
namespace ActsExamples {
    /// (Reconstructed) track parameters e.g. close to the vertex.
    using TrackParameters = ::Acts::BoundTrackParameters;
    /// Container of reconstructed track states for multiple tracks.
    using TrackParametersContainer = std::vector<TrackParameters>;

    using TrackContainer =
        Acts::TrackContainer<Acts::VectorTrackContainer,
                            Acts::VectorMultiTrajectory, std::shared_ptr>; // and const equiv
} // namespace ActsExamples

template <ACTS_CONCEPT(TrackContainerBackend) track_container_t,
         typename traj_t,
         template <typename> class holder_t = detail::RefHolder>
class TrackContainer {
    // stuff
};
```

PodioTrackContainer to the Rescue

<https://github.com/acts-project/acts/blob/v31.2.0/Plugins/Podio/edm.yml>

datatypes:

ActsPodioEdm::Track:

Description: "Reconstructed track"

Author : "Paul Gessinger, CERN"

Members:

- ActsPodioEdm::TrackInfo data // local parameters and covariance
- ActsPodioEdm::Surface referenceSurface // reference surface

ActsPodioEdm::TrackState:

Description: "Local state on a track"

Author : "Paul Gessinger, CERN"

Members:

- ActsPodioEdm::TrackStateInfo data // local information
- ActsPodioEdm::Surface referenceSurface // reference surface

PodioTrackContainer to the Rescue

<https://github.com/acts-project/acts/blob/v31.2.0/Plugins/Podio/edm.yml>

```
datatypes: // continued
```

```
  ActsPodioEdm::BoundParameters:
```

```
    Description: "Local track parameters"
```

```
    Author : "Paul Gessinger, CERN"
```

```
    Members:
```

```
      - ActsPodioEdm::BoundParametersInfo data // parameter information
```

How does Acts use ActsPodioEdm?

<https://github.com/acts-project/acts/blob/v31.2.0/Plugins/Podio/include/Acts/Plugins/Podio/PodioTrackContainer.hpp>

```
class MutablePodioTrackContainer : public PodioTrackContainerBase {
public:
    MutablePodioTrackContainer(const PodioUtil::ConversionHelper& helper)
        : PodioTrackContainerBase{helper},
          m_collection{std::make_unique<ActsPodioEdm::TrackCollection>()} { /*stuff*/ }
private:
    std::unique_ptr<ActsPodioEdm::TrackCollection> m_collection;
};

ACTS_STATIC_CHECK_CONCEPT(TrackContainerBackend, MutablePodioTrackContainer);
```

Can we use ActsPodioEdm?

Not so easily. Podio allows extension of data models only once. We extend edm4hep, so we cannot also extend ActsPodioEdm.

Can we copy the ActsPodioEdm structures into our data model? Yes, but we would still have to copy the PodioTrackContainer code as well, since that is tied to ActsPodioEdm.

Can we modify PodioTrackContainer to allow for various data models? Essentially we can upstream `PodioTrackContainer<T = ActsPodioEdm::Track>` which would then also allow `PodioTrackContainer<edm4eic::Track>`.

What are the Short Term Plans?

1. Let's start at the front end: avoid copying into ActsExamples structures for measurements and track parameters (not duplicating the ActsPodioEdm).
2. Create a local version of the Acts Podio plugin for development against our data model, temporarily housed in the EICrecon code base, but intended to upstream.
- 3.