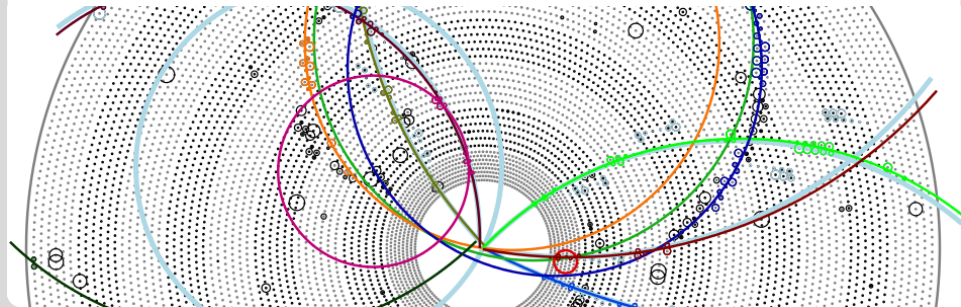![KIT logo - Karlsruher Institut für Technologie]

# RecoTrack Discussion

Biweekly Meeting.

Nils Braun │ 19.2.2016

KIT

# `RecoTrack`

## From F2F in Karlsruhe

- The `RecoTrack` should be the new dataobject in the tracking package which inherits from `genfit::Track`.
- It has therefore the same capabilities than the track and:
  - Relation-ready interface for the added hits (not reco hits).
  - `RecoHitInformation` object to save additional hit information (like RL-information in the CDC or sorting parameters).
  - Fitting preparation procedure build into the track (time-of-flight, AbsRep creation etc.)
  - Flag if last fit was successful.
- `genfit::TrackCandidate` objects can be build out of `RecoTrack` objects and vice versa.

# Proposal

- `RecoTrack` is derived from `RelationObject` and has a `genfit::Track` as a private member.
- In the beginning: Transformation functions for `genfit::TrackCands` and `genfit::Track`s.
- Addition of detector hits (not `TrackPoints`) by a simple-to-use interface. Internal "dirty" flag
- `TrackPoint`-Creation and Fitting with default parameters easy, but also possible with custom settings.
- Modules for all necessary steps provided, but all steps can also done in your modules.
- Only connection to `genfit::Track`: getter for `TrackPoints` and fit result/status.

# How does fitting work?

Internally:

1. Delete all `TrackPoints`, if the dirty flag is set.
2. Create `TrackPoints` for every related hit with the default setting or with configurable `MeasurementCreators`.
3. Fit the internal `genfit::Track` with the default settings or with a configurable `genfit::AbsFitter` instance.
4. Synchronize (the hit content) and the fit status.

Externally:

```
MeasurementAdder measurementAdder;
measurementAdder.setMeasurementCreatorsToDefaultSettings();
TrackFitter trackFitter;
trackFitter.setFitterToDefaultSettings();
// ...
measurementAdder.addMeasurements(recoTrack);
trackFitter.fit(recoTrack /*, trackRep */);
```

# How to add hits to a `RecoTrack`

- Hits-Adding is handled via relations to the `RecoTrack` and to a created `RecoHitInformation` object.
- The creation of these relation is handled internally and is not to be done by the user.
- There are convenience functions for the getters and setters for hit information and hit lists.
- The type of the hits is determined by a `typedef` in the `RecoHitInformation` class.

```
RecoTrack recoTrack;

recoTrack.addCDCHit(&cdcHit, sortingParameter);
recoTrack.addSVDHit(&svdHit, sortingParameter);

// Fitting etc.
```

# Discussion