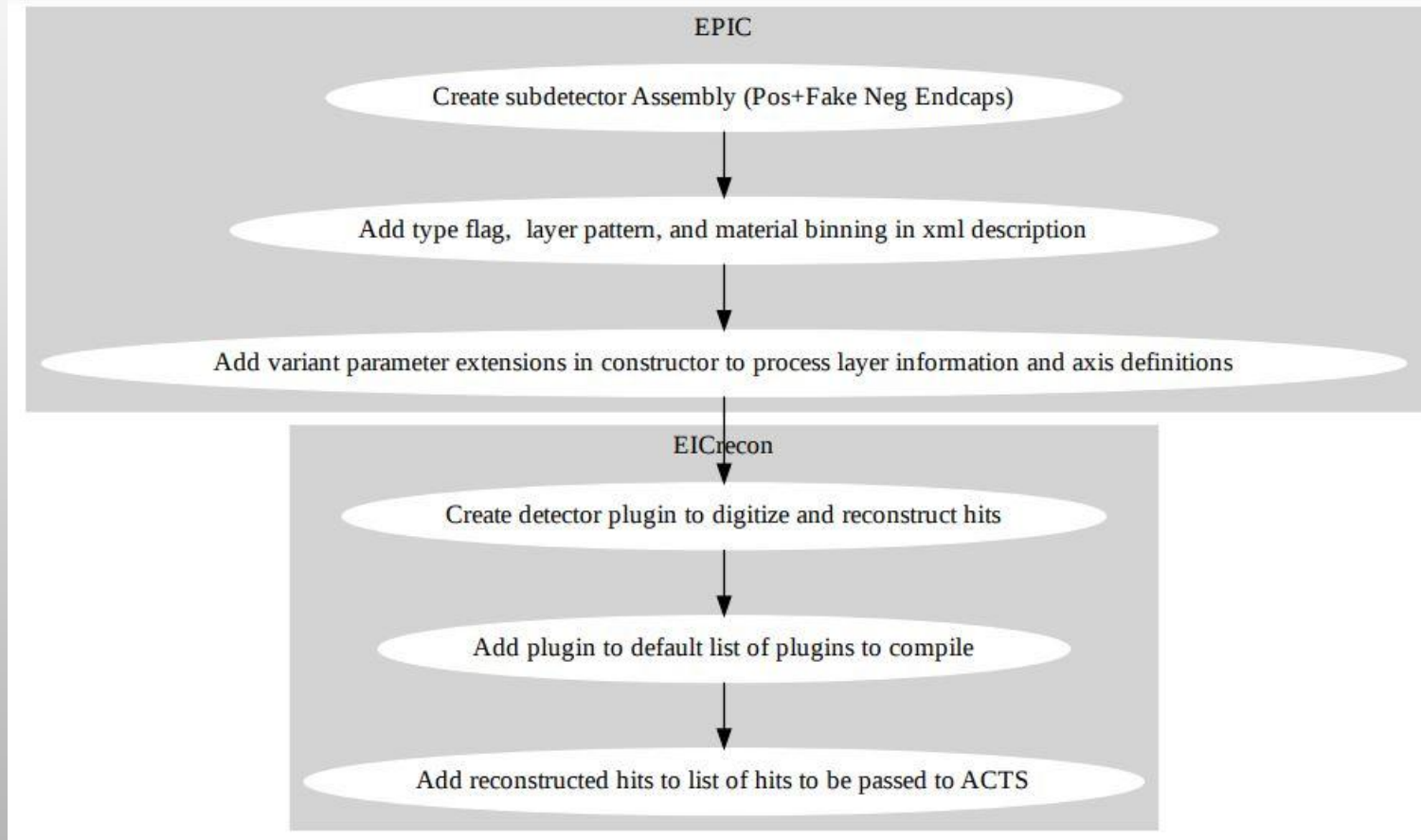


B0 Tracker Integration for Reconstruction

Sakib Rahman
EPIC-ACTS Joint Meeting
05 January 2023

Integrating the B0 Trackers into ACTS tracking for EPIC



Integrating the B0 Trackers into ACTS tracking for EPIC

Step 1:

Add B0 tracker
subsystem as a dd4hep
subdetector assembly

(positive endcap+fake
negative endcap each
with their own xml
description and
constructor)

**Not a limitation of
ACTS but of the
DD4hep translator.
ACTS is currently
adding a feature to
avoid the above
workaround of fake
companion.**

[compact/tracking/definitions.xml](#)

```
<detectors>
-----

  <detector id="B0TrackerSubAssembly_ID"
            name="B0TrackerSubAssembly"
            type="DD4hep_SubdetectorAssembly"
            vis="TrackerSubAssemblyVis">
    <composite name="B0Tracker"/>
    <composite name="B0TrackerCompanion"/>

  </detector>
-----
</detectors>
```

Integrating the B0 Trackers into ACTS tracking for EPIC

Step 2:
Set detector type
flag

B0Tracker.xml

```
<detectors>
  <detector
    id="B0Tracker_Station_1_ID"
    name="B0Tracker"
    type="ip6_B0Tracker"
    readout="B0TrackerHits"
    vis="FFTrackerVis">
    <type_flag type=
      "DetType_TRACKER"
      DetType_ENDCAP"/>
  </detector>
</detectors>
```

B0Tracker_geo.cpp

```
static Ref_t create_B0Tracker(Detector& description, xml_h e,
SensitiveDetector sens)
{
  xml_det_t      x_det      = e;
  DetElement     sdet(x_det.nameStr(), x_det.id());
  Volume         motherVol = description.pickMotherVolume(sdet);

  -----

  dd4hep::xml::setDetectorTypeFlag(x_det, sdet);
  auto &params = DD4hepDetectorHelper::ensureExtension
    <dd4hep::rec::VariantParameters>(sdet);

  for (xml_coll_t bmat(x_det, _Unicode(boundary_material));
       bmat; ++bmat) {
    xml_comp_t x_boundary_material = bmat;
    DD4hepDetectorHelper::xmlToProtoSurfaceMaterial(
      x_boundary_material, params,
      "boundary_material");
  }

  -----

}

DECLARE_DETELEMENT(ip6_B0Tracker, create_B0Tracker)
```

Integrating the B0 Trackers into ACTS tracking for EPIC

Step 3: Set layer parameters

B0Tracker.xml

```
<detectors>
  <detector
    id="B0Tracker_Station_1_ID"
    name="B0Tracker"...>

-----

    <layer id="4">
      <envelope rmin_tolerance=.../>
      <ring phi0 =...
        dphi =...
        r =...
        zstart = ...
        nmodules=...
        dz=...
        module="Module1" />
      <layer_material surface="inner"
        timing="binPhi.binR"
        bins0=...
        bins1=.../>
    </layer>

-----

  </detector>

-----

</detectors>
<plugins>
  <plugin name="DD4hep_ParametersPlugin">
    <argument value="B0Tracker"/>
    <argument value="layer_pattern:
      str=B0Tracker_layer\d"/>
  </plugin>
</plugins>
```

B0Tracker_geo.cpp

```
static Ref_t create_B0Tracker(Detector& description, xml_h e, SensitiveDetector
sens)
{
-----

  for (xml_coll_t li(x_det, _U(layer)); li; ++li) {
    xml_comp_t x_layer(li);
    string layer_name = det_name + std::string("_layer") +
      std::to_string(x_layer.id());
    xml_comp_t l_env = x_layer.child(_U(envelope));
    double layer_rmin_tolerance = l_env.attr<double>(_Unicode(rmin_tolerance));

-----

    DetElement layer_element(sdet, layer_name, x_layer.id());
    auto &layerParams = DD4hepDetectorHelper::ensureExtension
      <dd4hep::rec::VariantParameters>(layer_element);
    layerParams.set<double>("envelope_r_min", layer_rmin_tolerance);

-----

    for (xml_coll_t lmat(x_layer, _Unicode(layer_material)); lmat; ++lmat) {
      xml_comp_t x_layer_material = lmat;
      DD4hepDetectorHelper::xmlToProtoSurfaceMaterial(x_layer_material,
        layerParams, "layer_material");
    }
  }

-----
}
```

Integrating the B0 Trackers into ACTS tracking for EPIC

Step 4: Set axis definitions of sensitive elements

B0Tracker.xml

```
<detectors>
  <detector
    id="B0Tracker_Station_1_ID"
    name="B0Tracker"...>

-----

    <module name="SmallModule4"
      vis="FFTrackerModuleVis">
      <trd x1="B0TrackerMod1_x1/2.0"
        x2="B0TrackerMod4Small_x2/2.0"
        z="B0TrackerMod4Small_y/2"/>
      <module_component
        thickness="0.12*mm"
        material="CarbonFiber"
        vis="FFTrackerSupportVis" />
      <module_component
        thickness="0.3*mm"
        material="SiliconOxide"
        vis="FFTrackerSurfaceVis"
        sensitive="true"/>
    </module>

-----

  </detector>
</detectors>
```

B0Tracker_geo.cpp

```
static Ref_t create_B0Tracker(Detector& description, xml_h e, SensitiveDetector
sens)
{
-----

  for (xml_coll_t ri(x_layer, _U(ring)); ri; ++ri) {
    -----

    for (int k = 0; k < nmodules; ++k) {
      DetElement module(layer_element, m_base + "_pos", det_id);
      -----

      for (size_t ic = 0; ic < sensVols.size(); ++ic) {
        PlacedVolume sens_pv = sensVols[ic];
        DetElement comp_elt(module, sens_pv.volume().name(), mod_num);
        comp_elt.setPlacement(sens_pv);
        auto &params = DD4hepDetectorHelper::ensureExtension
          <dd4hep::rec::VariantParameters>(comp_elt);
        params.set<std::string>("axis_definitions", "XZY");
        volSurfacelist(comp_elt)->push_back(volplane_surfaces[m_nam][ic]);
      }
    }
  }
  -----

}
```

Integrating the B0 Trackers into ACTS tracking for EPIC

Add B0tracker detector plugin
to digitise and reconstruct
hits with detector info

```
void InitPlugin(JApplication *app) {  
    InitJANAPugin(app);  
    using namespace eicrecon;  
    app->Add(new JChainFactoryGeneratorT<SiliconTrackerDigi_factory>  
             ({"B0TrackerHits"}, "B0TrackerRawHits"));  
    app->Add(new JChainFactoryGeneratorT<TrackerHitReconstruction_factory>  
             ({"B0TrackerRawHits"}, "B0TrackerRecHits"));  
}
```

Step 5:
Activating the
tracker in
ElCrecon

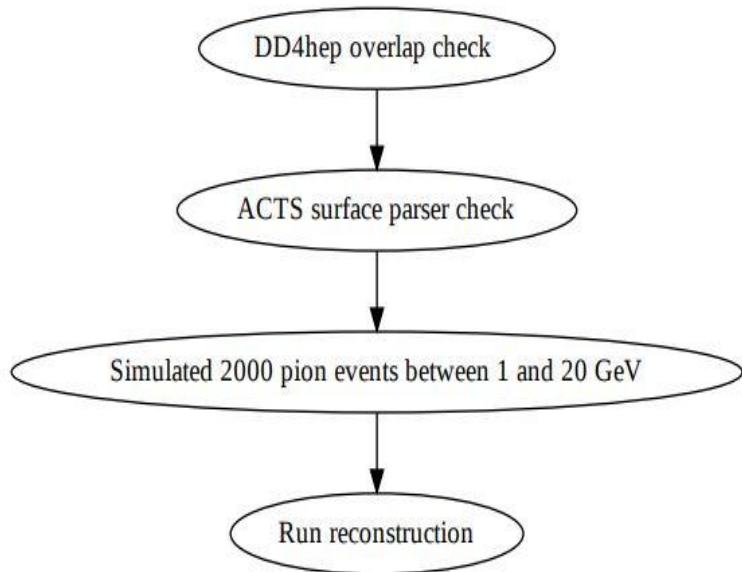
```
std::vector<std::string> EICRECON_DEFAULT_PLUGINS = {  
    ...  
    "B0TRK"  
    ...  
}
```

Add B0tracker plugin to default list
of plugins to activate in
src/utilities/eicrecon/eicrecon.cc

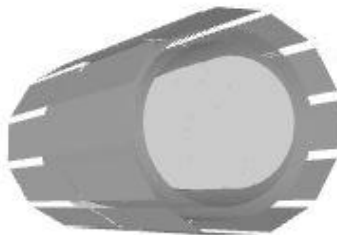
Add B0TrackerRecHits to be used in
src/global/tracking/tracking.cc

```
app->Add(new JChainFactoryGeneratorT<TrackerHitCollector_factory>(  
    {  
        "SiBarrelTrackerRecHits",  
        "SiBarrelVertexRecHits",  
        "SiEndcapTrackerRecHits",  
        "TOFBarrelRecHit",  
        "TOFEndcapRecHits",  
        "MPGDBarrelRecHits",  
        "MPGDDIRRecHits",  
        "B0TrackerRecHits"  
    },  
    "CentralTrackingRecHits"));
```

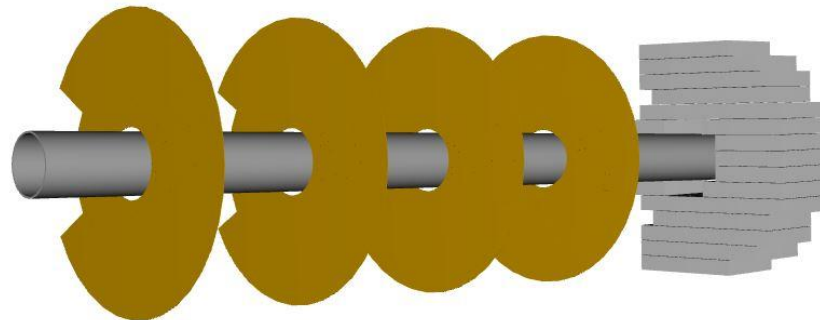
Test simulation and reconstruction chain



Central Region



B0 Trackers



Test simulation and reconstruction chain

```
ddsim -compactFile epic_tracking_only.xml -G -gun.particle "pi-" -gun.momentumMin  
"1*GeV" -gun.momentumMax "20*GeV" -gun.distribution "uniform" -N 2000 -outputFile  
sim_pi-_1GeV_20GeV_tracking_only.edm4hep.root
```

```
install/bin/run_eicrecon_reco_flags.py sim_pi-_1GeV_20GeV_tracking_only.edm4hep.root  
rec_result -Pplugins=dump_flags,janadot -Pdump_flags:json=tracking_only_flags.json
```

<https://github.com/eic/epic/pull/344>

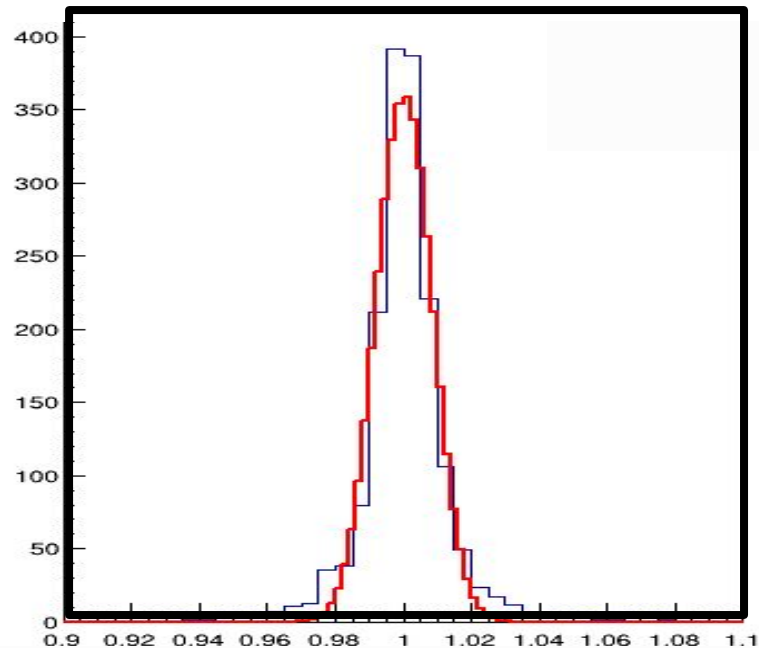
<https://github.com/eic/ElCrecon/pull/405/files>

Results

Overall reconstruction seems to produce reconstructed momentum fit with reasonable width but fails for events recording hits on B0

Cannot recognize hit to be on the surface associated with the hit cell id

ReconstructedZMomentum/TruthZMomentum



B0TrackerRechits

