ACTS DD4hep plugin

MuColl / Key4hep tracking meeting

Leonhard Reichenbach

2024-09-02

Clarifications

- Both DD4hep and ACTS have a plugin mechanism
- I will talk about a plugin in ACTS that is used to load DD4hep geometries :)
- I will first talk about the old plugin, then about the ACTS tracking geometry version 2 and 3

Geometry plugins in ACTS

- `GeoModel`: not exactly relevant for us
- `TGeo`: used by MuColl at the moment, 100s of lines (as far as I can see)
- `DD4hep`: the solution? Ideally only this

```
m_trackingGeo = Acts::convertDD4hepDetector(dd4hepGeo->world(), ...);
```

DD4hep plugin (old)

- Utilizes some information from DD4hep and a lot of `VariantParameters` added to the DD4hep geometry to parse the structure
- Under the hood has a thin wrapper around the TGeo plugin to convert the geometry after all the parts are arranged correctly
- Heavily used by the ACTS developers to load the OpenDataDetector, sadly also one of the only detectors that load

DD4hep plugin requirements (old)

- Positive endcap, negative endcap and barrel have to be 3 distinct detector elements with the same parent
- In particular if an endcap is positive or negative is determined by the z component of the placement. In `k4geo` this is most often 0, due to usage of assemblies at this level
- Layers of a sub-detector are selected using a RegEx attached as VariantParameter to the geometry
- Also some VariantParameters to assign material and envelopes but not as strictly required as the above

ACTS tracking geometry v2 (Blueprint)

- Newer mechanism to build a “layer-less” geometry
- Completely independent of the v1 geometry
- The last time I checked it could not do tracking yet...
- Tons of VariantParameters needed to describe everything
- ~550 extra lines of xml per sub-detector

ACTS tracking geometry v2 (Blueprint)

```
<plugin name="DD4hep_ParametersPlugin">
  <argument value="/world/OpenDataTracker/ShortStrips/ShortStripBarrel/
  <argument value="acts_volume: bool = true"/>
  <argument value="acts_volume_type: int = 3"/>
  <argument value="acts_volume_internals: bool = true"/>
  <argument value="acts_volume_internals_type: str = layer"/>
  <argument value="acts_volume_internals_measure: str = z,r"/>
  <argument value="acts_volume_internals_clearance: double = ss_b_clea
  <argument value="acts_surface_binning_dim: int = ss_b_sf_b"/>
  <argument value="acts_surface_binning_z_type: str = equidistant"/>
  <argument value="acts_surface_binning_z_autorange: bool = true"/>
  <argument value="acts_surface_binning_z_n: int = ss_b_sf_b_z"/>
  <argument value="acts_surface_binning_z_exp: int = ss_b_sf_e_z"/>
  <argument value="acts_surface_binning_phi_type: str = equidistant"/>
```

ACTS tracking geometry v2 (Blueprint)...

```
<argument value="acts_surface_binning_phi_n: int = ss_b0_sf_b_phi"/>
<argument value="acts_surface_binning_phi_exp: int = ss_b_sf_e_phi"/
<argument value="acts_portal_proto_material_2: bool = true"/>
<argument value="acts_portal_proto_material_2_binning_dim: int = 2"/
<argument value="acts_portal_proto_material_2_binning_z_type: str = 
<argument value="acts_portal_proto_material_2_binning_z_n: int = ss_
<argument value="acts_portal_proto_material_2_binning_z_autorange: bo
<argument value="acts_portal_proto_material_2_binning_phi_type: str =
<argument value="acts_portal_proto_material_2_binning_phi_n: int = ss
<argument value="acts_portal_proto_material_3: bool = true"/>
<argument value="acts_portal_proto_material_3_binning_dim: int = 2"/
<argument value="acts_portal_proto_material_3_binning_z_type: str = 
<argument value="acts_portal_proto_material_3_binning_z_autorange: bo
<argument value="acts_portal_proto_material_3_binning_z_n: int = ss_
```

ACTS tracking geometry v2 (Blueprint). . . . . .

```
    <argument value="acts_portal_proto_material_3_binning_phi_type: str =
    <argument value="acts_portal_proto_material_3_binning_phi_n: int = s
  </plugin>
```

- Needed for every individual layer
- Maybe not the most optimal solution?

ACTS tracking geometry v3

- WIP Attempt to take the best parts of v1 and v2 without all the chaos
- Driven by GeoModel plugin development

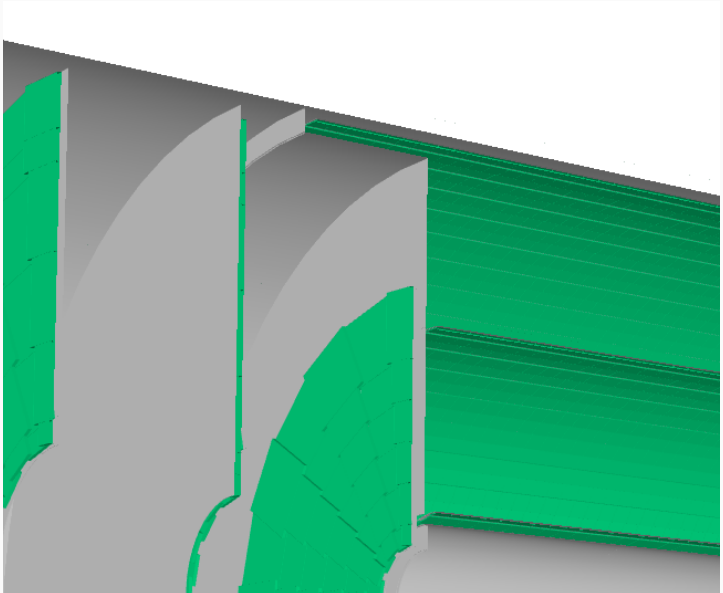
Status and plans for the future of the plugin

- Ideal goal: use as much of the existing information in DD4hep
- After all, we already have enough information to do tracking!
- After discussing with Paul Gessinger we agreed on the following:
  - For the beginning only use DD4hep information to arrange all parts of the detector correctly
  - I.e. parse the CellID encodings of the sensitive surfaces to identify which detector elements need to be converted to ACTS
  - Puzzle them together using the Protolayer functionality of ACTS
  - Let ACTS worry about everything else (material description will slightly differ)
  - Later on try to directly convert DDRec surfaces to ACTS surfaces

Obstacles identified so far

- CLD geometry loads too slow for efficient development (fixed in `CLD_o2_v07`)
- Needs to be validated against OpenDataDetector, but
  - ODD encoding scheme is different for every sub-detector and has no sides (fix in preparation)
  - ODD surfaces added twice to SurfaceManager... (fix submitted)
  - ODD surfaces point in changing directions (to be fixed for part 2, but easy)
- CLD passive surfaces all have `cellID == 0` as it was not needed for readout
- CLD geometry has alternating active and passive surfaces that have different parents so the grouping into a common volume needs some work

Obstacles identified so far



Outlook

- Maybe only a couple more days added to have a working prototype.
- Unfortunately I have only negative time left to work on this until the end of October. . .