CEPCSW Integration

Jin Zhang

Status of CEPCSW integration

- Environment - Lcg98, CEPCSW
- Acts Components
  - Core, DigitizationPlugin, IdentificationPlugin, DD4hepPlugin
- Building Acts Geometry
- Building Acts Geometry using DD4hep
  - Demon Detector
  - Fullsilicon Detector
- Current related work && plan to be discussed

Building ACTS Geometry

- An example to build ACTS tracking geometry with Acts::GenericDetector
- Building example directory: Detectors/GenericExample

```
[zhangjin@lxslc709 src]$ ls
BuildGenericDetector.cpp  GenericDetector.hpp~        GenericDetectorElement.hpp  GenericExample.hpp  ProtoLayerCreatorT.hpp
BuildGenericDetector.hpp  GenericDetectorElement.cpp  GenericExample.cpp          LayerBuilderT.hpp
```

- In GenericExample.cpp :

```
using TrackingGeometryPtr = std::shared_ptr<const Acts::TrackingGeometry>;
 /// Return the generic detector
 TrackingGeometryPtr gGeometry
        = Generic::buildDetector<DetectorElement>(nominalContext,
                          detectorStore,
                          buildLevel,
                          std::move(mdecorator),
                          buildProto,
                          surfaceLogLevel,
                          layerLogLevel,
                          volumeLogLevel);
```

- ./run Examples/options/GenericActs.py

```
13:01:19    CylinderVolu   VERBOSE   [ end ] return newly created container : { { { { BeamPipe::Barrel | { Pixel::NegativeEndcap
 | Pixel::Barrel | Pixel::PositiveEndcap } } | PST::Barrel } | { SStrip::NegativeEndcap | SStrip::Barrel | SStrip::PositiveEndca
p } } | { LStrip::NegativeEndcap | LStrip::Barrel | LStrip::PositiveEndcap } }
finish building
ApplicationMgr       INFO Application Manager Stopped successfully
EventLoopMgr         INFO Histograms converted successfully according to request.
ApplicationMgr       INFO Application Manager Finalized successfully
ApplicationMgr       INFO Application Manager Terminated successfully
[zhangjin@lxslc709 CEPCSW]$
```

Building ACTS Geometry using DD4hep

- An example to build dd4hep detector with ActsExtension, then to convert to Acts TrackingGeometry
- A Demonstrator detector built in Detector/Demon in DD4hep_generate_rootmap

```
[zhangjin@lxslc709 src]$ ls
DemonstratorBarrel_geo.cpp  DemonstratorBeamPipe_geo.cpp
```

- ActsGeometry building example: Reconstruction/Acts/Detectors/DD4hepExample
  - Building DD4hepGeometry with GeomSvc
  - Using Acts::Converter to translate into ACTS Geometry
- ./run Examples/options/DD4hepActs.py

```
// Set the tracking geometry
auto trackingGeometry
        = std::move(Acts::convertDD4hepDetector(m_dd4hep,
                                    logLevel,
                                    bTypePhi,
                                    bTypeR,
                                    bTypeZ,
                                    envelopeR,
                                    envelopeZ,
                                    defaultLayerThickness,
                                    sortDetectors,
                                    gctx,
                                    mdecorator));
```

```
13:13:55    DD4hepConver    INFO      Translating DD4hep geometry into Acts geometry
13:13:55    DD4hepConver    INFO      Translating DD4hep sub detector: beampipe
13:13:55    D2A_Logger      VERBOSE   Processing detector element:  beampipe
```

```
13:13:55    D2A_CVH         VERBOSE   [ end ] return newly created container : { beampipe::Barrel | { beampipe::fGap | barrel::Ba
rrel | beampipe::sGap } }
ApplicationMgr       INFO Application Manager Stopped successfully
EventLoopMgr         INFO Histograms converted successfully according to request.
ApplicationMgr       INFO Application Manager Finalized successfully
ApplicationMgr       INFO Application Manager Terminated successfully
```

Building ACTS Geometry: FullSilicon

- FullSilicon detector : Detector/FullSilicon
- ./run Examples/options/FullSilicon.py

```
14:20:53    D2A_CVH        VERBOSE   [ end ] return newly created container : { BeamPipe::Barrel | { BeamPipe::fGap | { SOT_EOT_
4::NegativeEndcap | { { SOT_EOT_4::fGap | { SOT_EOT_3::NegativeEndcap | { { SOT_EOT_3::fGap | { SOT_EOT_2::NegativeEndcap | { {
SOT_EOT_2::fGap | { SOT_EOT_1::NegativeEndcap | { { SOT_EOT_1::fGap | { FST_VXD_EIT::NegativeEndcap | FST_VXD_EIT::Barrel | FST_
VXD_EIT::PositiveEndcap } | SOT_EOT_1::sGap } | SOT_EOT_1::Barrel } | SOT_EOT_1::PositiveEndcap } | SOT_EOT_2::sGap } | SOT_EOT_
2::Barrel } | SOT_EOT_2::PositiveEndcap } | SOT_EOT_3::sGap } | SOT_EOT_3::Barrel } | SOT_EOT_3::PositiveEndcap } | SOT_EOT_4::s
Gap } | SOT_EOT_4::Barrel } | SOT_EOT_4::PositiveEndcap } | BeamPipe::sGap } }
ApplicationMgr       INFO Application Manager Stopped successfully
EventLoopMgr         INFO Histograms converted successfully according to request.
ApplicationMgr       INFO Application Manager Finalized successfully
ApplicationMgr       INFO Application Manager Terminated successfully
[zhangjin@lxslc709 CEPCSW]$
```

Current related work && plan to be discussed

- Issues
  - First sight on some Acts modules - current to do
    - Propagation - available for debugging, independent from CEPCSW EDM
    - Material Studies - available, tools of Geant4 can be used , But Material Mapping need Json, which is missing in CEPCSW
    - Fast sim
      - FATRAS modules are missing in lcg98 and CEPCSW
      - Also need using **EDM** in CEPCSW
    - Kalman studies : requires Fast Sim
    - Further - Geant4 full sim
  - Geometry version
    - In CEPCSW, example is DemonDetector which is just one layer
    - In previous acts-framework, we have CDR detector (1)baseline (2)FullSi , in CEPCSW, also some different detectors
  - Performance validation : root file output and Algorithm parameter definition