Gaudi-Marlin-Processor Wrapper update

key4hep meeting

Plácido Fernández Declara

May 19, 2020

CERN

GMP Wrapper

- The GMP Wrapper project aims to smoothly bring Marlin functionality to Gaudi framework.
    - First steps are creating interfaces (wraps) around Marlin *Processors* using Gaudi *Algorithms*.
    - This keeps the current base source code working while using Gaudi framework.
    - Then different pieces of Marlin Processors can be ported progressively to replace functionality.
- Some bugs and configuration issues were fixed, a manual (`README.md`) was added with instructions on compiling, configuring, running and testing.
- This process allows for cleanup, modernization and optimization while keeping the functionality.

GMP Wrapper dependencies

GMP Wrapper can be built against an iLCSoft installation + Gaudi. Main dependencies:

- **Gaudi**: to wrap Marlin processors and run the algorithms.
- **Marlin**: to run the underlying processors
- **LCIO**: Event Data Model input/output

Other dependencies:

- **ROOT**
- **Boost**

GMP Wrapper GitHub: `https://github.com/andresailer/GMP` (Integrated in key4hep: `https://github.com/key4hep/`)

GMP Wrapper configuration and running

Configuring and running the wrapper is done as in Gaudi, through a Python file:

- An algorithm list is filled with wrapped Marlin Processors.
- Processors parameters are defined for each instance, defining the Marlin processor to load and list of parameters and values
  - Converter for Marlin XML configuration files exists

On algorithm initialization of a Marlin Processor, `MARLIN_DLL` environment variable is used to load the necessary libraries.

GMP configuration example

```
digiVxd = MarlinProcessorWrapper("VXDBarrelDigitiser")
digiVxd.OutputLevel = DEBUG
digiVxd.ProcessorType = "DDPlanarDigiProcessor"
digiVxd.Parameters = [
    "SubDetectorName", "Vertex", END_TAG,
    "IsStrip", "false", END_TAG,
    "ResolutionU", "0.003", "0.003", "0.003", "0.003", "0.003", "0.003", END_TAG,
    "ResolutionV", "0.003", "0.003", "0.003", "0.003", "0.003", "0.003", END_TAG,
    "SimTrackHitCollectionName", "VertexBarrelCollection", END_TAG,
    "SimTrkHitRelCollection", "VXDTrackerHitRelations", END_TAG,
    "TrackerHitCollectionName", "VXDTrackerHits", END_TAG,
    "Verbosity" , "DEBUG", END_TAG,]
algList.append(digiVxd)
```

Testing

Added testing with `ctest`:

- Simple test that runs some Marlin Processors: AidaProcessor -> InitDD4hep -> VXDBarrelDigitiser
- `muon.slcio` is used for input, without hits.
- Second test generates an input file with `ddsim`
- It runs a similar list of algorithms with actual hits
- Output checks for regex with `INFO Application Manager Terminated successfully`

```
    ddsim \
        --steeringFile $ILCSOFT/ClicPerformance/HEAD/clicConfig/clic_steer.py \
        --inputFiles $ILCSOFT/ClicPerformance/HEAD/Tests/yyxyev_000.stdhep -N 4 \
        --compactFile $ILCSOFT/lcgeo/HEAD/CLIC/compact/CLIC_o3_v14/CLIC_o3_v14.xml \
        --outputFile $GMP_tests_DIR/inputFiles/testSimulation.slcio
```

Integration into key4hep

- Its status is now more usable for the general public
- CI/CD is in the way relying on the tests
- GaudiMarlinWrapper and where to put it in key4hep?