SoLID simulation thoughts

Zhiwen Zhao

2015/04/02

general

- My impression so far is no existing simulation framework will fit our need as it is
- We need to choose one with good potential to develop and tailer to our need , especially linked to our choice of reconstruction and calibration

GEMC development

- Mauri is planning to make GEMC development open formally, which means
  - Working group meeting
  - emaillist (gemc_software@jlab.org)
  - Core members (main developers from CLAS12 and SoLID)
  - Work together to plan roadmap, implement feature and fix bug and ensure continuity.
  - Convert repo from SVN to github for better collaborating, bug reporting, feature request

Event generator

- External, pipe line by file
  - Pro: independence, flexible, little overhead when adopting new generator, run once and simulate many times
  - Con: better fix format early on to ensure compatibility
- Internal, pipeline with mem
  - Pro: format can involve with simulation
  - con: more overhead to adopt new, not efficient for many jobs or repeating jobs

database

- Can’t avoid it for last scale detectors for calibration and survey data
- The real question is if there’s a way to avoid overhead to use it when develop locally
- But maybe it’s possible to minimize the overhead if set up correctly (mirror server, CCDB?)

Detector definition

- SoLID still have many different configuration
- And things are far from fixed now
- Hardcoded it in source code would create many exe files and a lot headache came with it
- GEMC definition is just like vanilla geant4, only take necessary part outside of source code
- The real question is how organize and track changes

arrows with different color mean different interface