Full Simulation status

Andrea Di Simone
INFN Tor Vergata

Background frames

- Feedback was given at SLAC concerning frames produced during last production
  - Main suggestion was to implement some filtering in order to reduce the number of particles passed t the fast sim
    - Keep only particles which are likely to produce hits
    - -200cm<|z|<250cm
    - Energy for photons > 8MeV
  - Implemented, committed, tagged
  - Note: this is all hard-coded, no configurability, very specific to bg frame production

Code validation

- After implementing latest features, a small validation pre-production was run
  - Result was not encouraging
    - 60% of jobs was failing
- This lead to a (painful) debugging exercise, involving several developers
  - Not going into the details now: at the end, it looks like during one of the commits a real G4 bug was uncovered
  - Its effects are masked by the step limitation in the final focus
    - we can safely run productions, provided we keep that setting (ON by default)
- Apart from the specific case, I believe the lesson here is about software validation
  - We should agree on a minimal set of tests that every developer is requested to perform (and pass) before committing his/her code

Code validation (2)

- As a seed for the discussion, I would suggest:
  - In case of gdml changes, run the geometry test before and after the modification
    - check that no NEW clashes are introduced
  - For all the rest, simulate at least 500 beamstrahlung events, and at least 5000 single particles (electrons or hadrons, depending on the detector affected)
    - All jobs must succeed
- This should be run ideally at every commit by the developers
  - Should we provide helper scripts for job submission?
- To be complemented by tests with higher statistics when tagging/releasing
  - To be run by whoever tags/releases

Profiling

- I typically add a comment about profiling being crucial in the todo list of my talks
- This time it is something different
  - I am running a Bruno job (single electrons) through cachegrind
  - I was hoping to have some results already for this meeting, but apparently I underestimated the slowness of cachegrind
- I will inspect the output as soon as I have it, and start contacting relevant developers in case I find bottlenecks