Getting Started with Simulation in NuWa

Brett Viren

Physics Department



Daya Bay Simulation Phone, 2008/02/04

Contents

1. Overview & Status
2. Goals fo Getting Started
3. DetDesc Geometry
4. GenTools Kinematics Generation
5. DetSim Detector Simulation
6. Summary

Goals of Getting Started

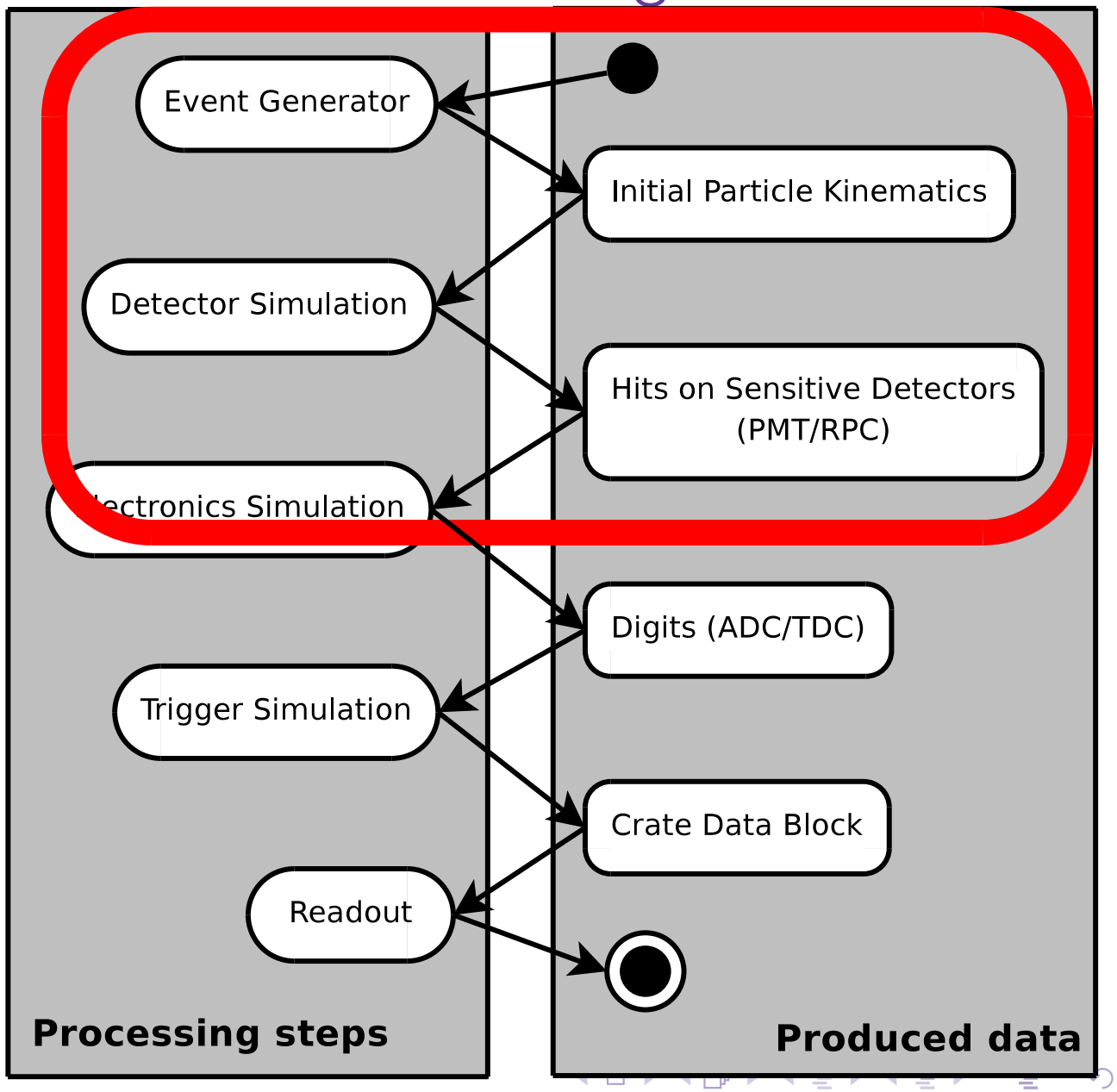
Goals I think people should reach after getting started:

- Understand what works now and what is still a work-in-progress.
- Learn how to exercise what does work now.
- Learn how to configure the simulation stages.
- Learn details of the C++ classes and the XML geometry files.

I hope you can get involved in finishing, fixing and improving things so we can start doing real studies!

Context: Simulation has well defined stages.

- Reminder of general simulation organization.
- My focus is now on the first two stages.
- There is work in progress on the second two stages by others.



Status: Geometry

This includes the description of detector volumes, materials, properties and structure and populating the Transient Detector Store (TDS).

- Packages: `Detector/XmlDetDesc` & `Detector/DetDescVis`
- Done: basic Site, Pool and PMT (simplified version of Tak Pui's 3 hemisphere design)
- Partly Done: OpenInventor visualization (via `quanjing.py`), G4 visualization possible, RayTracer not working (optical properties?).
- To do: PMT placements, ADs, RPCs and **lots of detail**.
- To do: write down XML file organization conventions.

Status: Kinematics Generation

Generators create the initial particle kinematics and fill the Transient Event Store (TES) with the resulting `HepMC::GenEvent` objects.

- Package: `Simulation/GenTools`
- Done: basic framework classes and test scripts.
- Done: "legacy" hepevt generators supported.
- Done: parameterized particle "gun" generator done.
- Done: "positioner" and "timererator" generators done (place event in space and time.
- In progress: Better integration of "legacy" generators (Karen)

Status: Detector Simulation

The detector simulation tracks particles and produces hits. There are two:

`Simulation/DetSim` (bv):

- GiGa/Geant4 based
- Uses `GenTools` output as its input
- Uses `DetDesc` geometry
- Works now, but no TES filler yet (leverage `MCTruthMaker`)

`Simulation/SimuAlg` (xingtao/xinchun):

- G4dyb/Geant4 based
- Handles generators directly, doesn't yet use `DetDesc`/TES for input
- Uses "legacy" hard coded geometry
- Works now and writes data to TES (via `MCTruthMaker`)

Xinchun, Xingtao and myself will make them compatible with each other
$\Rightarrow$ comparison studies between G4dyb (`SimuAlg`) and GiGa (`DetSim`).

Getting Started, Generalities

- A Python configuration and test script exists in `PackageName/python/packagename.py`.
- Most "should" be able to just run
- You will likely need to "`source setup.sh`" in the `cmt/` directory of the packages mentioned before using their scripts.
- Development is moving quickly so things may break from time-to-time.
- When this slows down I will write a manual, but until then, there is always the source code to read!...

Getting Started with DetDesc Geometry

1. Need to generate pool geometry by hand. Use:
   `Detector/XmlDetDesc/python/XmlDetDescGen/Pool/gen.py`
   (this needs to be added to the build process)
2. Run simple test using:
   `Detector/XmlDetDesc/python/xmldetdesc.py`
3. Note the dump of the Transient Detector Store (TDS)
4. Visualize the geometry using:
   `Detector/DetDescVis/python/quanjing.py`
5. Examine the XML files in `DDDB/`
6. Read and understand `xmldetdesc.py`
7. Reference for XML schema:
   http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/DetDesc/Documents/lhcbdtd.pdf

Getting Started with GenTools

1. You can run a basic test with: `Simulation/GenTools/python/gentools.py`
   - It calls Co60.exe, fills TES with results, runs an algorithm that pull results out of TES and dumps them to screen.
   - You will likely need to first build `Generators/RadioActivity/Co60`.
2. Read and understand `gentools.py`
3. Understand how the C++ classes in `src/` work.
4. See DocDB 1729 and https://wiki.bnl.gov/dayabay/index.php?title=GaudiTool

Getting Started with DetSim

1. You can run a basic test with: `Simulation/DetSim/python/detsim.py`
2. Observe messages about kinematics generated and G4Events being created.
3. Try also `pmtbox.py`.
4. Try also `visdet.py` for G4 visualization - but this is still "expert" level.

DetSim is still the "freshest" part of the code, so don't be surprised if you find problems.

Summary

- Much of a Gaudi-based simulation exists now.
- Much work still to do, but interested people can start helping now.
- Send any problems to the offline or simulation mailing lists and not just to me. This way we all can learn.