DetDesc and DetSim: Detector Simulation and Description

Brett Viren

Physics Department



Daya Bay Offline Video Meeting, 2008/03/19

Contents

1. Detector Description (DetDesc)
   - Overview
   - Validation
   - Examples
2. Detector Simulation (DetSim)
   - Overview
   - Geant4 Interface
3. Current Status
   - Detector Description
   - Physics Lists
   - Kinematics Input
   - G4Event Production and TES Filling Status
4. Summary & Comment

Overview of DetDesc

What is DetDesc?

- C++ classes/objects for in-memory representation:
  - Describe materials, shapes, volumes and full logical and abbreviated physical containment hierarchies.
  - Convertable to Geant4 geometry objects; provides additional features (eg. offsets from ideal geometry)
  - Gaudi-aware: objects are stored in the TDS.
  - Direct visualization via `quanjing.py`.
- An XML schema for in-file representation:
  - Supports parallel development of sub-detector descriptions.
  - Flexible referencing mechanism for sub-elements
  - Some other features: parameters, multi-placements, user-data, extensible detector elements.

Previously seen:

"Daya Bay Detector Description", Dec 2007 tutorial, DocDB #1579.

"DetDesc vs VGM/AGDD", Oct 2007, DocDB #1405.

3 Major parts: Geometry, Materials & Structure

Geometry Full logical/physical volume hierarchy

- In TDS @ `/dd/Geometry/`
- Complete description for simulation purposes

Structure Hierarchy of important (physical) volumes

- In TDS @ `/dd/Structure/`
- Hierarchy of "bookmarks" to important physical volumes in Geometry.
- Used to apply alignment offsets and for reconstruction

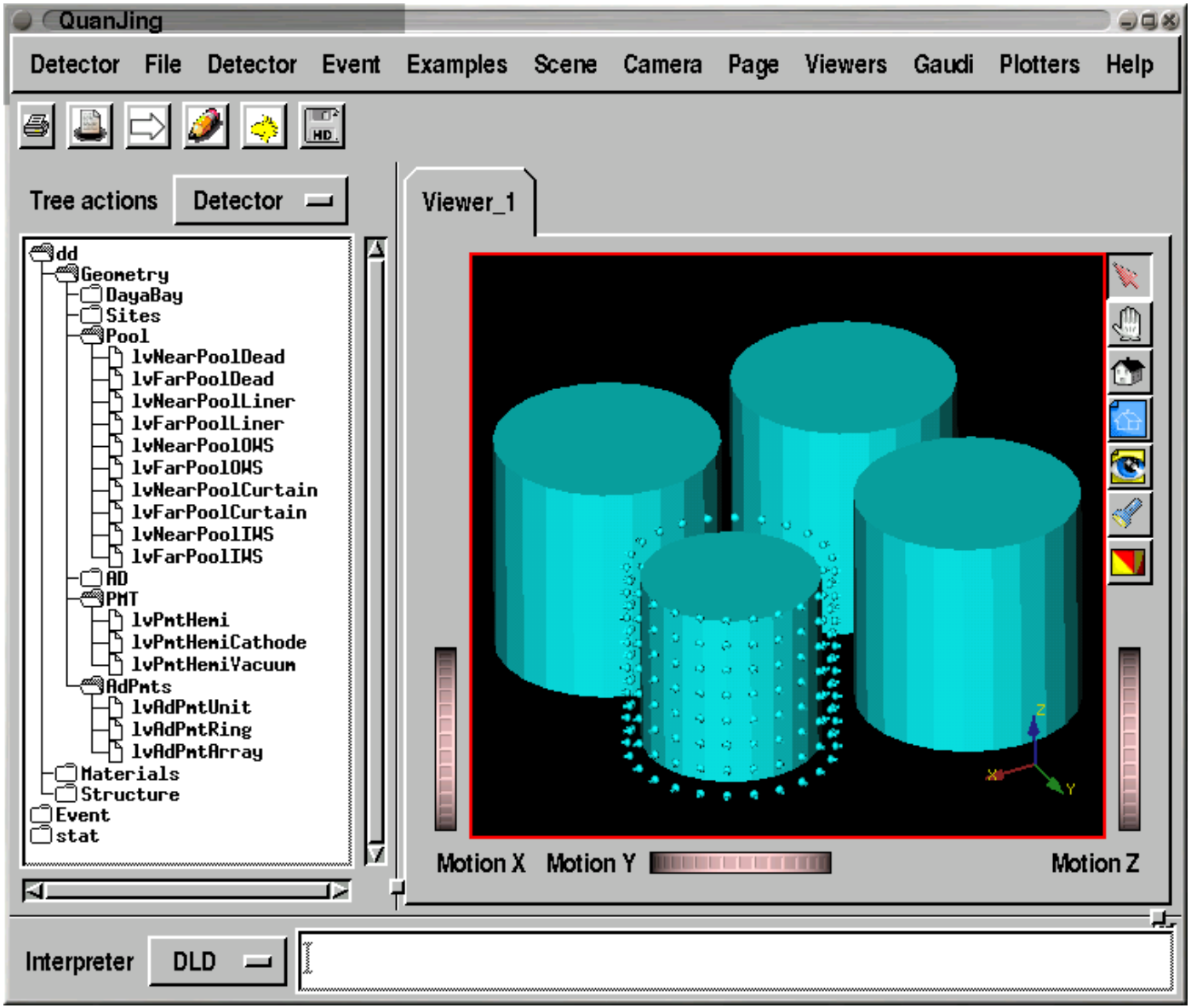
Materials Materials and their optical properties

- In TDS @ `/dd/Materials/`
- Material, elements, isotopes, properties.

Geometry validation

- Simple parse/dump XML syntax checking, basic geometry checks (`dump.py`)
- Direct (no G4) density "X-rays" of geometry (`VolumeCheckAlg`).
- Direct (no G4) visualization and inspection (`quanjing.py`)
- Geant4 visualization after conversion.

Direct visualization: `quanjing.py`



- Inspect TDS & TES
- Rotate, pan, zoom
- Interrogate volumes
- Delete outer to see inner volumes
- Transparency & color effects
- GUI built from simple XML
- Easy to plug in user code
- Can be basis for event display

A few examples...

1. AD oil cylinder with PMT array placement
2. The PMT model
3. Creating the AD PMT array

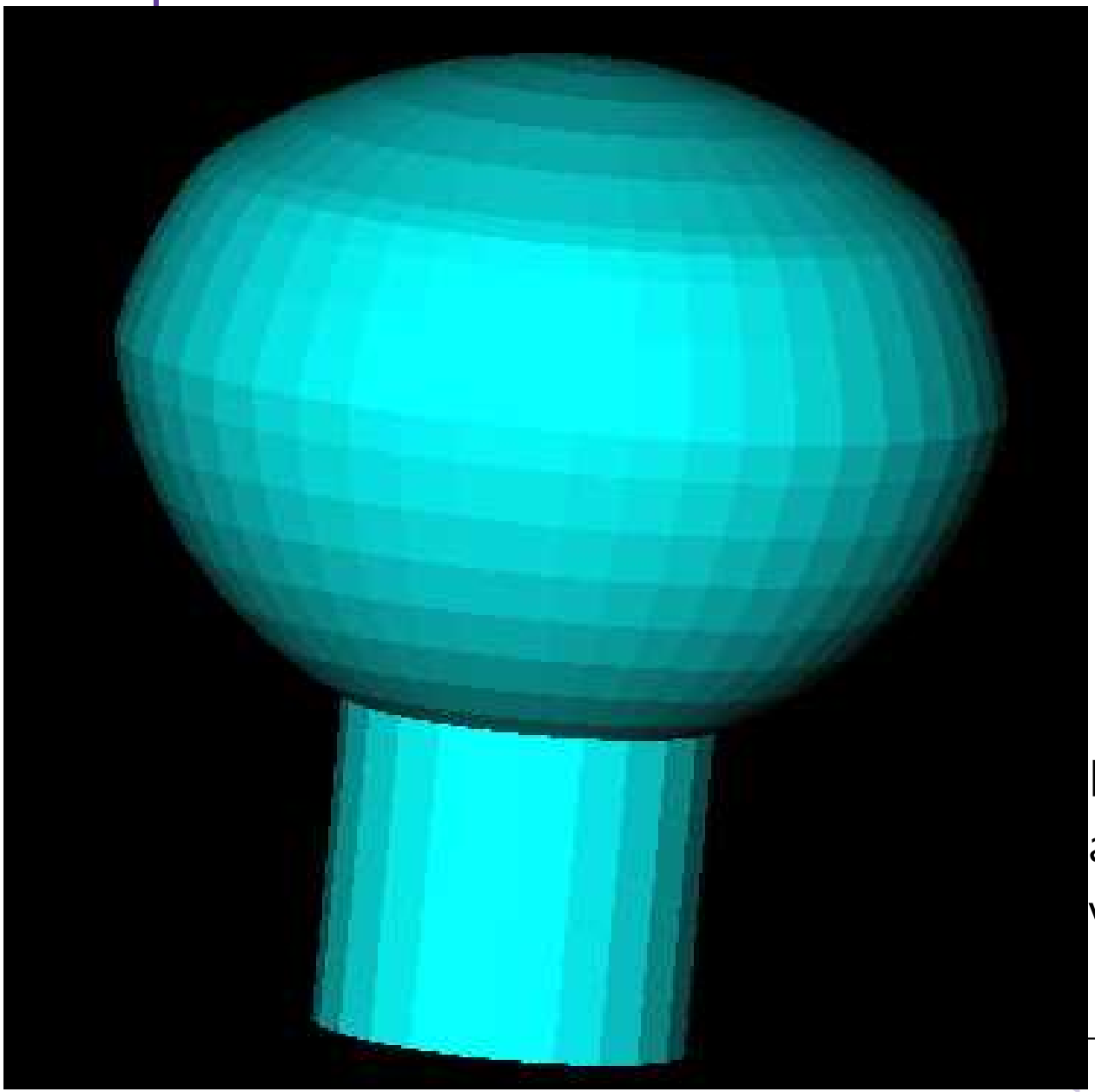
Example 1: AD Oil Cylinder + PMT Placements

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DDDB SYSTEM "../DTD/geometry.dtd"
  [<!ENTITY ADParameters SYSTEM "parameters.xml">]>
<DDDB>
&ADParameters; <!-- hard-coded or derived parameters -->

<logvol name="lvOIL" material="MineralOil">
  <!-- the shape: -->
  <tubs name="oil" sizeZ="ADoilHeight"
              outerRadius="ADoilRadius" />

  <!-- any physical daughter volumes: -->
  <physvol name="pvOAV"
          logvol="/dd/Geometry/AD/lvOAV"/>
  <physvol name="pvAdPmtArray"
          logvol="/dd/Geometry/AdPmts/lvAdPmtArray"/>
</logvol>
</DDDB>
```

Example 2: PMT



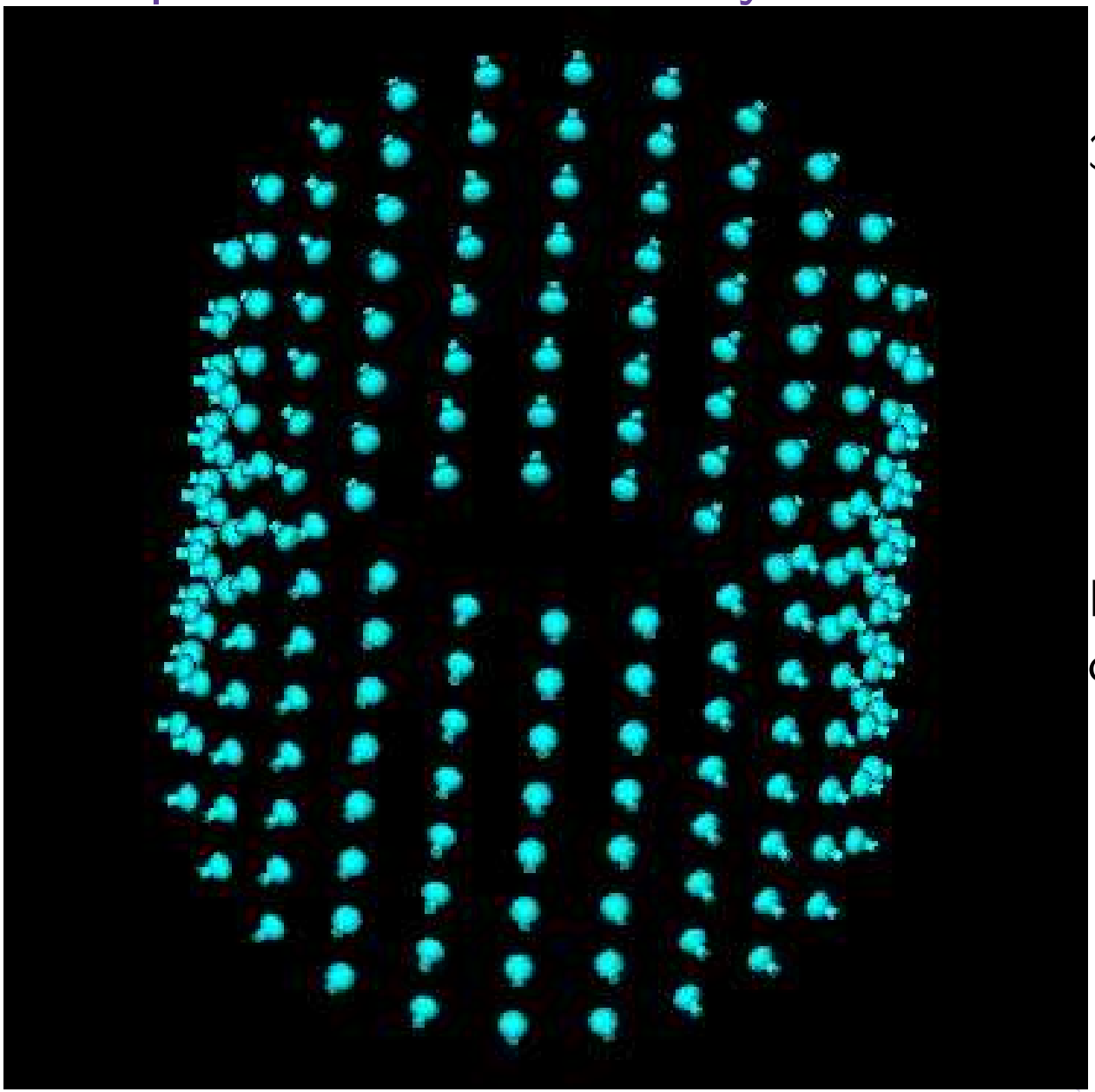
1. Tak-Pui's model[a], union of three hemispheres
2. Pyrex, photocathode and vacuum volumes
3. Photocathode / sensitive detector connection given in XML

Note: polygons are just an artifact of this particular visualization program.

[a] DocDB #1596

```
<logvol name="lvPmtHemi" material="Pyrex">
  <union name="pmt-hemi">
    <intersection name="pmt-hemi-glass-bulb">
      <sphere name="pmt-hemi-face-glass"
              outerRadius="PmtHemiFaceROC"/>
      <sphere name="pmt-hemi-top-glass"
              outerRadius="PmtHemiBellyROC"/>
      <posXYZ z="PmtHemiFaceOff-PmtHemiBellyOff"/>
     <sphere name="pmt-hemi-bot-glass"
              outerRadius="PmtHemiBellyROC"/>
      <posXYZ z="PmtHemiFaceOff+PmtHemiBellyOff"/>
    </intersection>
    <tubs name="pmt-hemi-base"
          sizeZ="PmtHemiGlassBaseLength"
          outerRadius="PmtHemiGlassBaseRadius"/>
    <posXYZ z="-0.5*PmtHemiGlassBaseLength"/>
  </union>
  <physvol name="pvPmtHemiCathode"
           logvol="/dd/Geometry/PMT/lvPmtHemiCathode"/>
  <physvol name="pvPmtHemiVacuum"
           logvol="/dd/Geometry/PMT/lvPmtHemiVacuum"/>
</logvol>
```

Example 3: AD PMT Array



3 steps:

1. Position one PMT at bottom of tank
2. Copy to make ring of 24 PMTs
3. Copy ring 8 times

Result can be placed as one volume.

```
<logvol name="lvAdPmtUnit">                    <!-- Step 1 -->
  <physvol name="pvAdPmtUnit"
           logvol="/dd/Geometry/PMT/lvPmtHemi">
    <posXYZ x="AdPmtRadialPos"
            z="-0.5*(AdPmtNrings-1)*AdPmtZsep"/>
    <rotXYZ rotY="-90*degree" />
  </physvol>  </logvol>

<logvol name="lvAdPmtRing">                    <!-- Step 2 -->
  <paramphysvol number="AdPmtNperRing">
    <physvol name="pvAdPmtRing"
             logvol="/dd/Geometry/AdPmts/lvAdPmtUnit" />
    <posXYZ/>
    <rotXYZ rotZ="AdPmtAngularSep" />
  </paramphysvol>  </logvol>

<logvol name="lvAdPmtArray">                   <!-- Step 3 -->
  <paramphysvol number="AdPmtNrings">
    <physvol name="pvAdPmtArray"
             logvol="/dd/Geometry/AdPmts/lvAdPmtRing"/>
    <posXYZ z="AdPmtZsep"/>
  </paramphysvol>  </logvol>
```

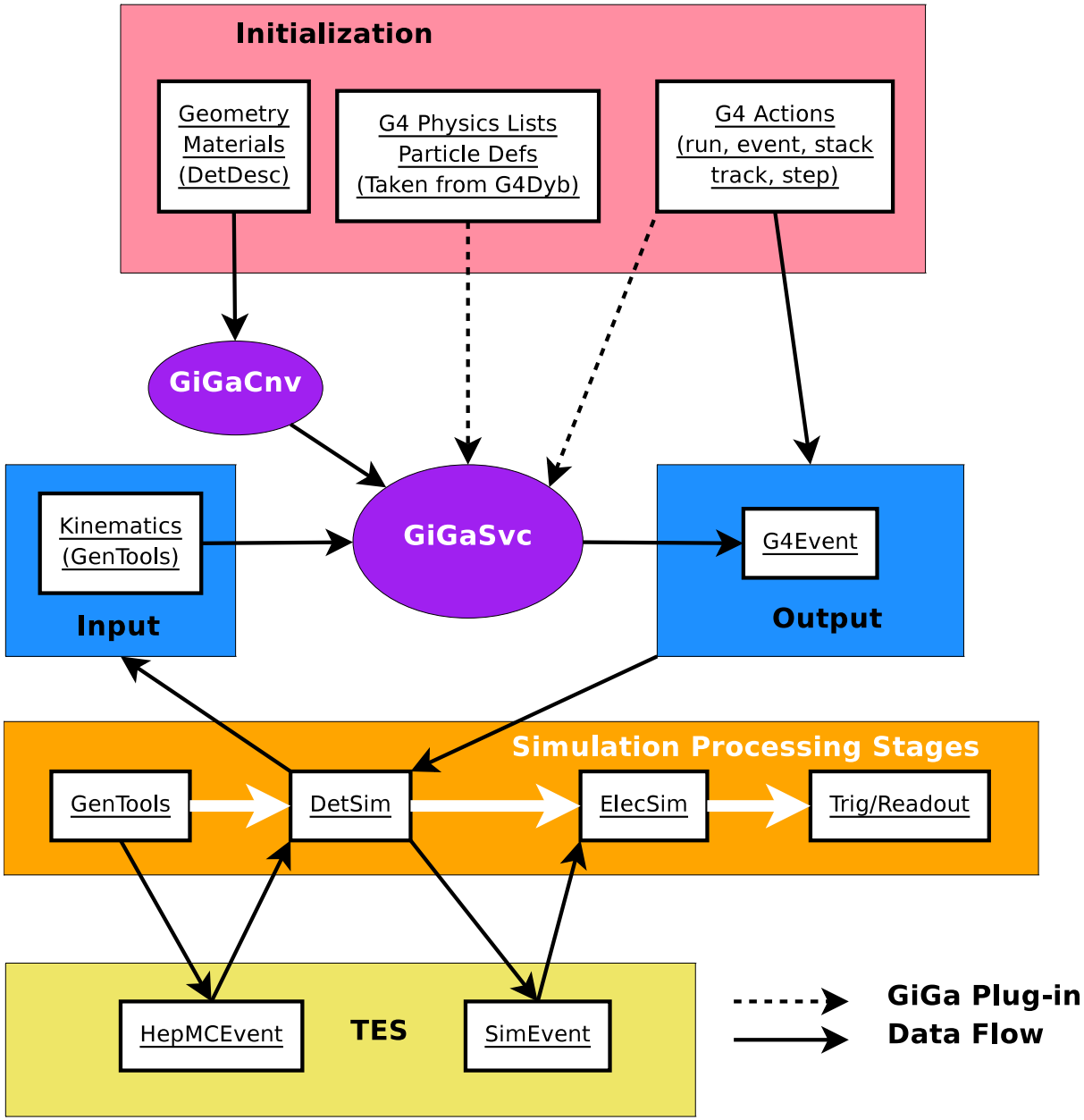
Overview of DetSim

What is DetSim?

- A Geant4 based simulation of particles tracked through Daya Bay detectors.
- Uses DetDesc as source of geometry and materials
- Retains the parts of G4dyb that **matter** to Daya Bay physics:
  - Generators (not in DetSim, proper, moved to GenTools[1])
  - Physics lists
  - Customized Cherenkov and Scintillator processes
  - Discards: geometry and data readout classes (bulk of G4dyb)
- Feeds GenTool's kinematics from TES into Geant4
- Pulls out Geant4 events and stores hit data into TES
- Collects trajectory and unobservable statistical data

[1] "GenTools, Generators in Gaudi", DocDB #1729

Interface to Geant4



- DetDesc → G4 geometry
- PhysList classes from G4dyb
- Action classes for unobservable statistics & trajectory recording
- Kine in, G4 data out
- DetSim algs interface between Kine & GiGa/G4 and TES

What is GiGa?

- Relatively thin glue between Gaudi and Geant4.
  - Simplified for the "user-executer".
  - Geant4 still fully exposed to the "user-programmer".
    - ★ Allows for all the usual Geant4 hacking like goes on with G4dyb.
    - ★ Organizes this hacking: one user doesn't fight another user's code.
    - ★ There is very little "GiGa stuff" to know.
- Manages Geant4's main loop
  - Same functionality as SimuAlg's G4SvcRunManager.
- Provides DetDesc $\longrightarrow$ G4 geometry object conversion.
- Run time configuration:
  - Which Actions, PhysicsLists or Geometry to use
  - What type/amount of data to record

How does a user work with GiGa?

Normally, a user needs to use, or slightly modify, default run-time configuration and geometry. Experts might:

- Modify which PhysicsLists to include via run-time config, or develop their own custom ones to include new physics.
- Develop new, custom physics processes (Ceren & Scint already taken from G4dyb)
- Write custom Run/Event/Track/Step/Stack Actions to record special information or implement special physics. G4dyb type quantities available by default.
- Modify the amount and type of data to record via run-time config.
- Modify, or radically extend/change, the geometry. No need to write or compile C++.
- Set up special simulations using code that is common with main simulation. Beyond new Geometry, little C++ devel needed. Example use: table-top PMT collection efficiency experiment modeling.

Detector Description Status

- **Sites** located and oriented as per 2/07 Yellow River drawings: done
  - Needs confirmation against latest drawings.
- Far and Near **halls** (per YR): done
- Far and Near water **pool** major structures: done
  - Needs PMT placements (Kevin[2] in progress) and AD tables
- **AD** major structure and PMT placements: done
  - Placement details being settled, tank-external structures needed
- DetDesc $\longrightarrow$ G4 **conversion** tested, works: done
- Basic **validation** procedures: done
  - Overlap checking and G4 visualizations still experimental
  - Direct comparison to G4dyb geometry not yet considered
- No **RPC**
- No details beyond what is in G4dyb; no Reactors

[2] Less than a week of effort, already an expert. "It is clear and easy to use."

Physics Lists Status

- Basic and Optical Physics lists taken from G4dyb: done
  - The others still to be copied in (trivial).
- Cerenkov process taken from G4dyb: done
  - Scintillation process still needs to be copied in (trivial)

Kinematics Input Status

- GenTools wrapper for legacy generator programs: done
  - Improvement using through native C++ interface (Karen): in progress
- Generic generators:
  - Particle "gun", various random distributions: done
  - Event time setting based on average an rate: done
  - Event vertex setting, various random distributions: done
- Working as input to DetDesc: done
  - SimuAlg needs modifying to do same: in progress(?)
  - XingTao & Xinchun agree to do this (still?)

G4Event Production and TES Filling Status

- G4Event production from kinematics: done
- Accessing hit information in sensitive detector (PMT only): done
- Writing of Hits to TES: in progress
  - Some delay for needed DataModel reorganization and TES design (time window implementation): both in progress, near completion
  - Careful thought needed for recording trajectories and non-observable statistics: idea stage.
  - SimuAlg needs modifying to write consistent format to make comparison studies possible.

Summary

Summary:

- A detector description (DetDesc) and simulation (DetSim) running in the context of Gaudi **exists now**.
- Basic implementation and testing is **largely done**.
- Expert sub-detector effort is needed to add details to the geometry (you know who you are).
- There are tasks for people of various skill levels contribute.
- Some low level issues are still being finished.
- User and programmer documentation needed (some stuff in wiki).

Personal Comments

- We have agreed to use Gaudi.
- I **have** spent time familiarizing myself with the issues and **do** have experience with what I'm talking about. Spreading misinformation is not doing any service to Daya Bay.
- DetDesc/DetSim is an **existing** and **largely working** Gaudi-based geometry and detector simulation.
- There is still work to be done and I could use help from **collaborators**.
- The proposal to start work on a **non-existent, redundant and non-Gaudi** geometry system at this point will:
  - Distract potential contributors from helping to finish DetDesc/DetSim as well as other open issues (eg. SimuAlg compatibility).
  - Disillusion the primary offline developer from continuing to make any effort and question if Daya Bay can be a functional collaboration.
  - Divide the collaboration in to two analysis groups.
- The resulting problems from wasting time on a redundant system when the existing system is essentially done will be cause for grave concerns by funding agencies.