LArG4 Refactoring and other changes

Why, how, and various animals

Status and plans

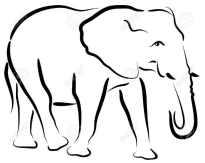
William Seligman

21-Nov-2017

The idea and concepts for the LArG4 changes primarily come from Hans Wenzel; I'm just doing some of the coding. Any errors of fact are mine, not Hans'.

Key

Speed improvements (we hope)



Caution: slow down and think

Potential memory issues

LArG4

Geant4 tracks particles through the LAr in discrete steps.

Each step deposits energy in two forms (that are relevant to this discussion):

- Ionization
- Scintillation

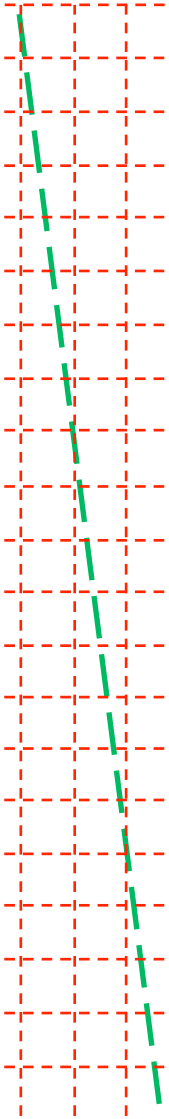
Ionization:

- Converted into number of electrons.
- Electrons are grouped into clusters.
- Clusters are drifted onto the wires (channels) of the planes.

Scintillation:

- Compute number of optical photons.
- Photons are checked against a library of pre-computed tables to see if they hit an optical detector.

LArG4: Current



The step size limited via voxels.

There are two voxel “worlds”:

charged particles = 0.3mm cubes (1/10th wire spacing)

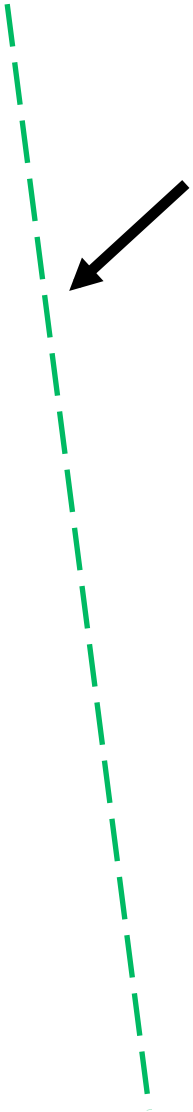
neutral particles = ~(5cm, 5cm, 3cm) in (x,y,z) for uB

(The actual sizes from .fcl files, and are specified separately in x,y,z.)

Problem: The step lengths along a track were not even, which created some non-physical effects.

I set up the voxel system back in 2008 for reasons that turned out not to be relevant, so…

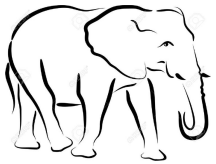
LArG4: Changes



Step sizes will come from Geant4's StepLimiter mechanism, assuring even step lengths along the track (except at the end of a track).

Again, the step lengths will come from .fcl files. For charged particles, it will start at 0.3mm.

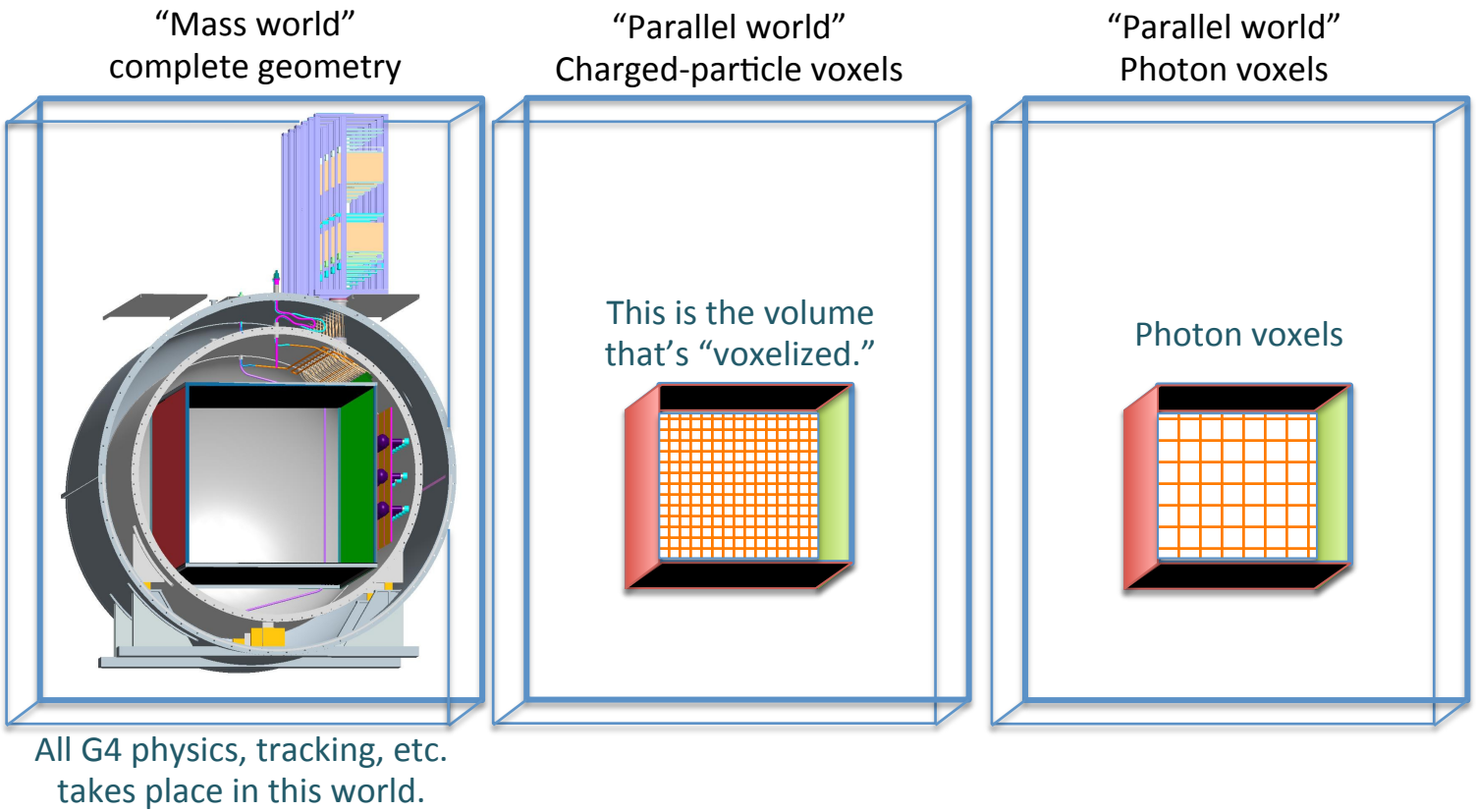
The voxel geometry will be replaced a single sensitive detector for both charged and neutral particles. Where the E-field is zero, there will be only scintillation.



There's still a class in Utilites called LArVoxelList. We haven't written voxels as a data product in years (see sim::SimChannel later in this talk), but there might be a program somewhere that still relies on the SimChannel->voxel conversion utility.

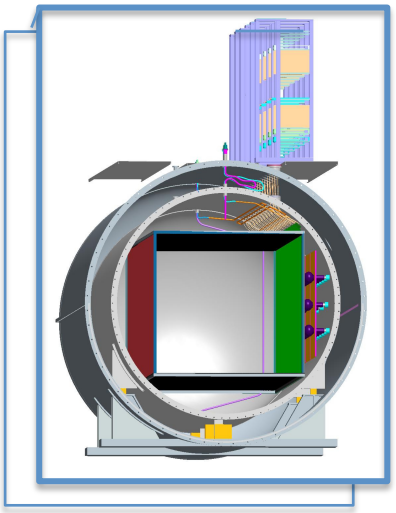
LArG4: Original

Geant4 parallel worlds in LArG4

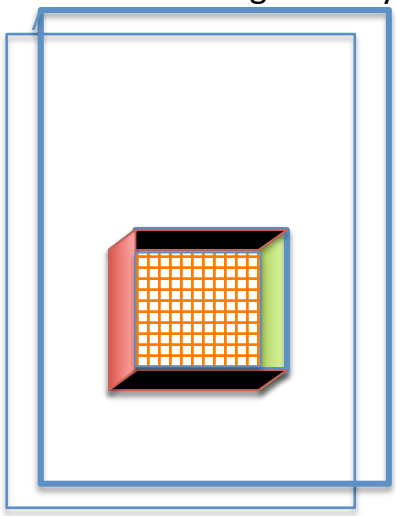


LArG4: Current

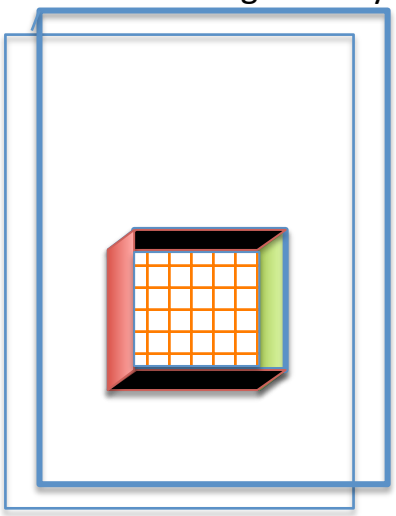
"Mass world"
complete geometry

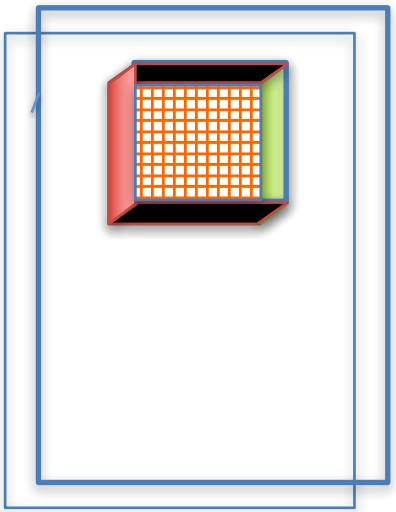


"Parallel world"
wire readout geometry

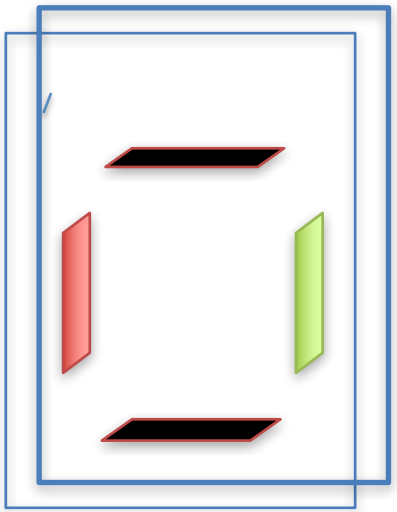


"Parallel world"
PMT readout geometry





One parallel world for
each LAr TPC



Another parallel world
just for AuxDets

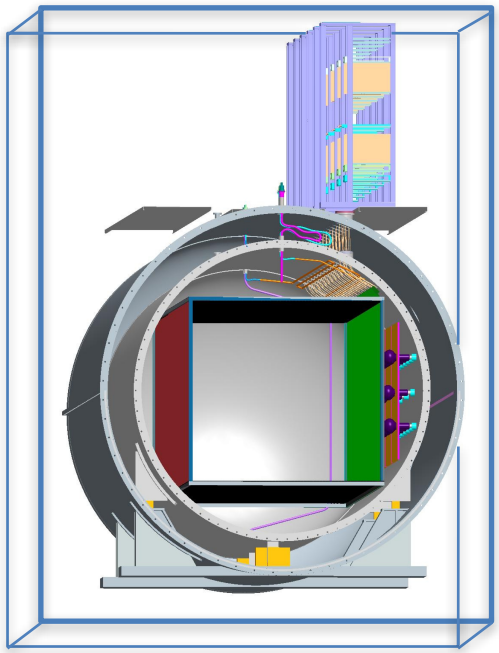
There may
be more.
G4 speed
issues?

Earth-2, Earth-X,
Earth-19, …

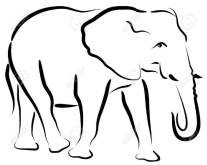
LArG4: Changes

The Geant4 parallel worlds will go away. They were never needed for the AuxDets, and without the voxels parallel worlds are no longer needed.

Just the Geant4 geometry



Crisis on Infinite Earths…



Some experiments do some tricks with hits in AuxDets. If they're based solely on volume name, there won't be any problem.

LArG4: Physics Lists

The amount of Ionization and Scintillation from each step is based on a physics list.

Reminder: A “physics list” is a collection of processes and models that Geant4 uses. While one can specify each process individually, in practice we use the physics lists that are packaged along with Geant4.

LArG4: Current

The amount of Ionization and Scintillation from each step is based on a physics list.

Ionization:

LArG4 uses the QGSP_BIC physics list, which is somewhat old and optimized more for ATLAS than LAr.

Scintillation:

Contrary to the previous slide, the optical photons come from a custom LArG4 routine which overrides the G4 physics list with individual processes.

LArG4: Changes

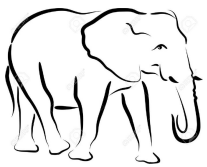
Geant4 packaged physics lists have improved over the past few years. Hans Wenzel has moved the control of physics lists to G4Helper in nutools. LArG4 will no longer replace physics processes that are already present in Geant4.

G4Helper now uses the new Geant4 extendable physics list factory. This allows use of all the reference physics lists provided by Geant4 which then can be extended by specific physics processes; e.g.

FTFP_BERT+OPTICAL+STEPLIMIT+NEUTRONLIMIT

specifies the FTFP_BERT reference physics list and then adds optical processes, step limiter, and a time limit on neutrons.

The optical physics process now allows for the option not to put photons on the stack, which is what we want for LArG4 (see Accumulating Photons slide).



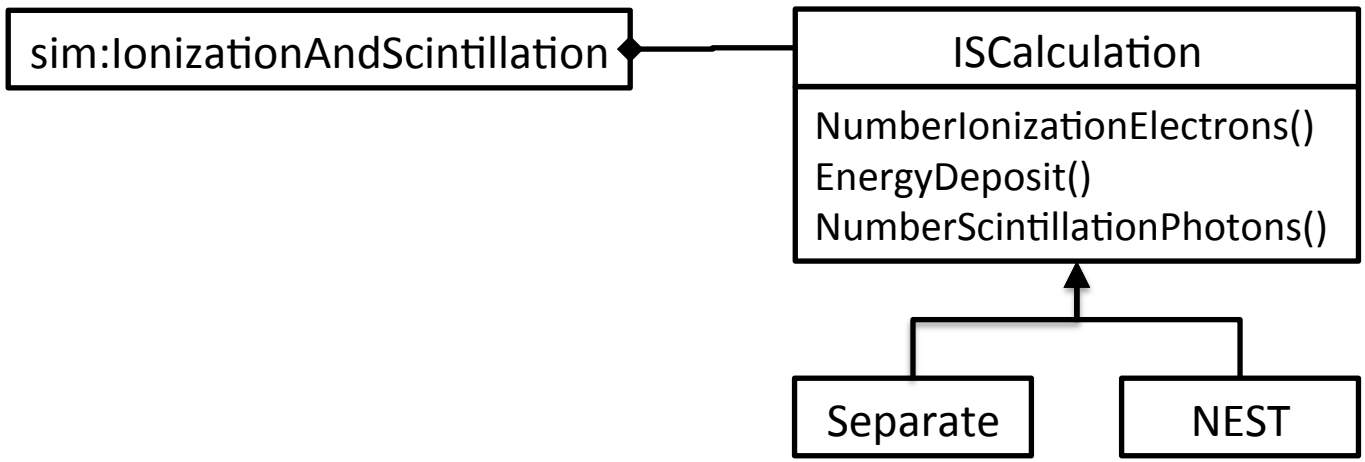
G4Helper in nutools is the logical place for controlling physics lists, but it may confuse folks who want to do custom studies; e.g., turn off delta rays. (This is a documentation issue.)

LArG4: I&S

Physics List -> Ionization and Scintillation
The "I&S" calculation is what we do with these results of the physics list.

I&S:
Ionization -> number of electrons (we do this; it would take too long for Geant4 to calculate this for us)
Scintillation -> number of photons (not a tautology; see next slide)

The I&S calculation is called for each step in the simulation. It's implemented using the strategy design pattern:



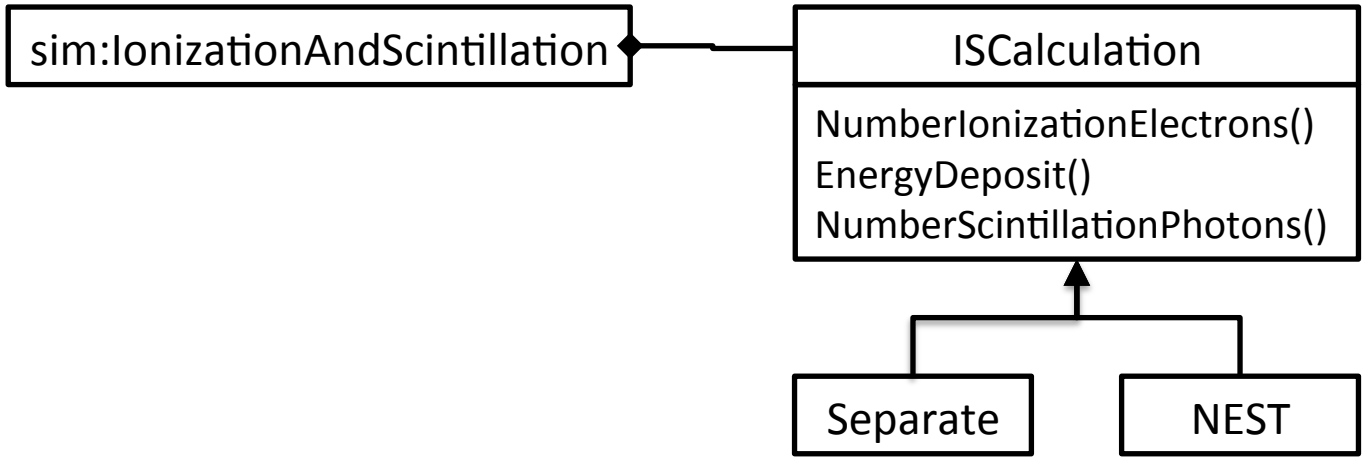
LArG4: Current I&S Options

”Separate”:

- Ionization and Scintillation yields are ‘uncorrelated’
  - (Once we shift to using G4’s physics lists, they will be)
- Uses recombination model for Ionization->number of electrons
  - Optional box model
- Number of photons from physics list
  - Optional saturation

“NEST”:

- Correlated Ionization and Scintillation yield (recalculates both)
- Complex (i.e., slow) calculation

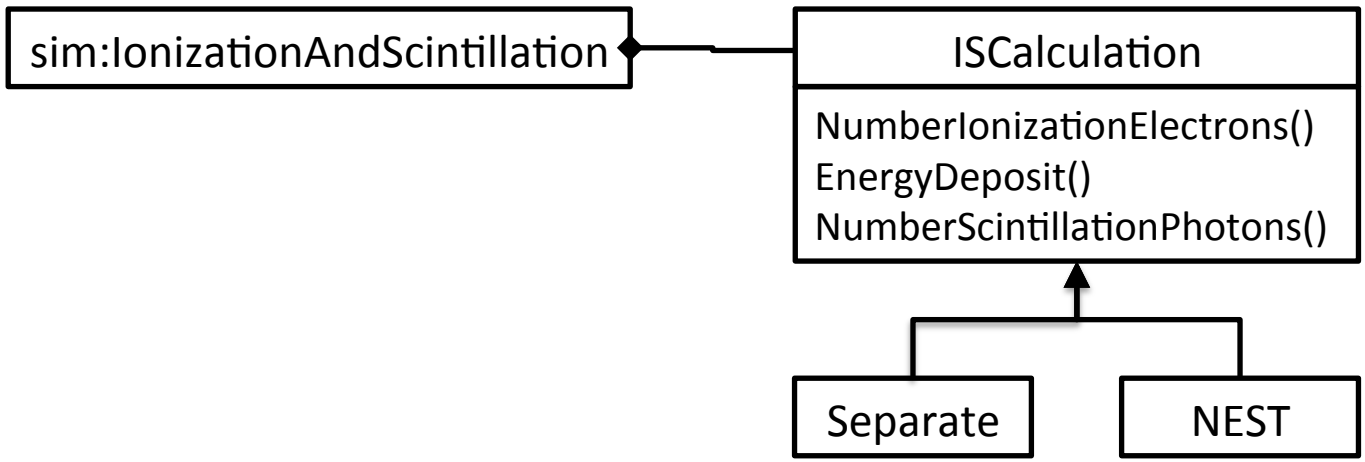


LArG4: Abandoned Changes

The original plan was to move the I&S calculation outside of LArG4. This would allow groups with alternate I&S models to test them without rerunning Geant4.

Unfortunately, NEST depends heavily on Geant4 routines. Someone who understands the calculation might be able to rewrite the NEST code, but it looks like a non-trivial task.

(Yes, this is disappointing.)



Accumulating photons

The story so far:

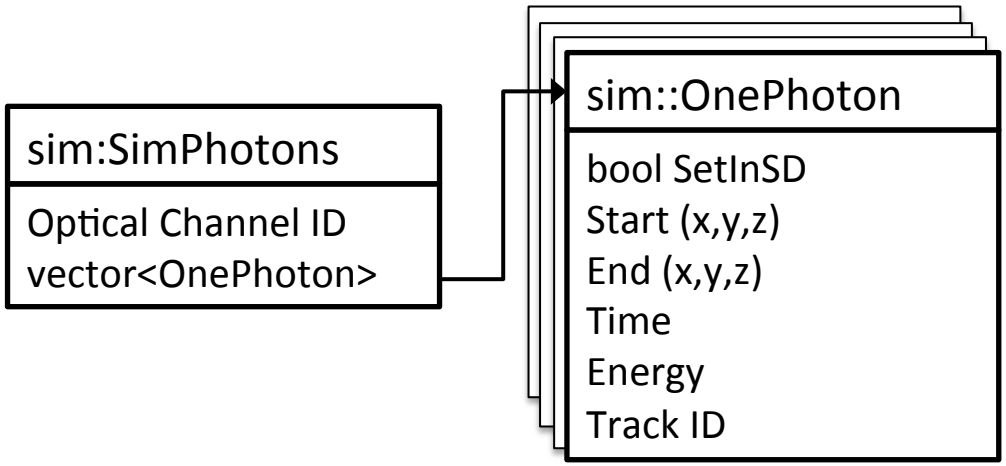
Ionization -> number of electrons

Scintillation -> number of optical photons

Number of optical photons:

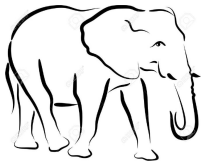
Each scintillation photon with a given (x,y,z) has a probability of generating a response in a given optical detector. Instead of having G4 propagate every photon (time consuming), we take them off the stack and use a "photon library" previously created by a dedicated job.

If a photon would generate a response in an optical detector, a OnePhoton object is added to the SimPhotons data product.



Photon Library

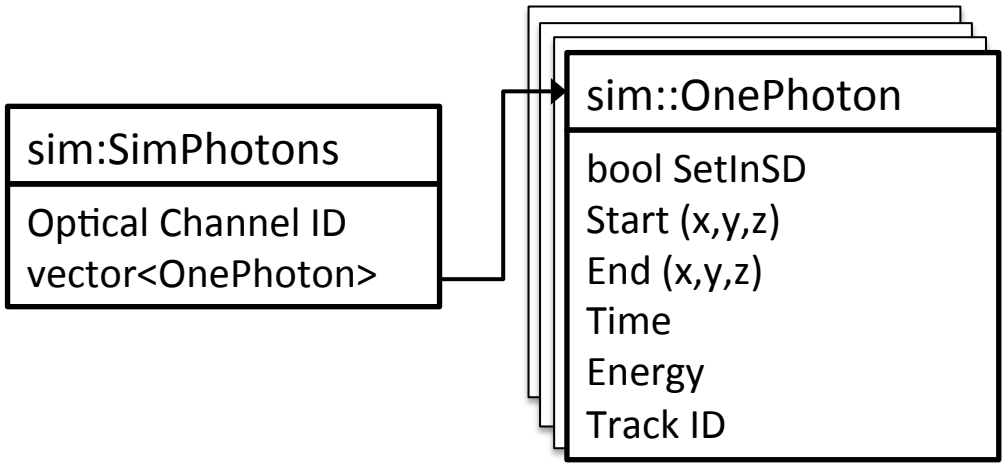
The photon library only needs to be regenerated when the detector geometry is changed or there's a major change to the optical physics list. It's a time-consuming job taking several CPU days.



With the changes Hans is making to the GDML files (e.g., including reflection properties) and to the physics lists, the photon library should be regenerated.

We're making lots of changes to LArG4. We have to check that the photon-library generation procedure still works.

Photon voxels won't vanish as a concept. We use them as binning for the photon library.

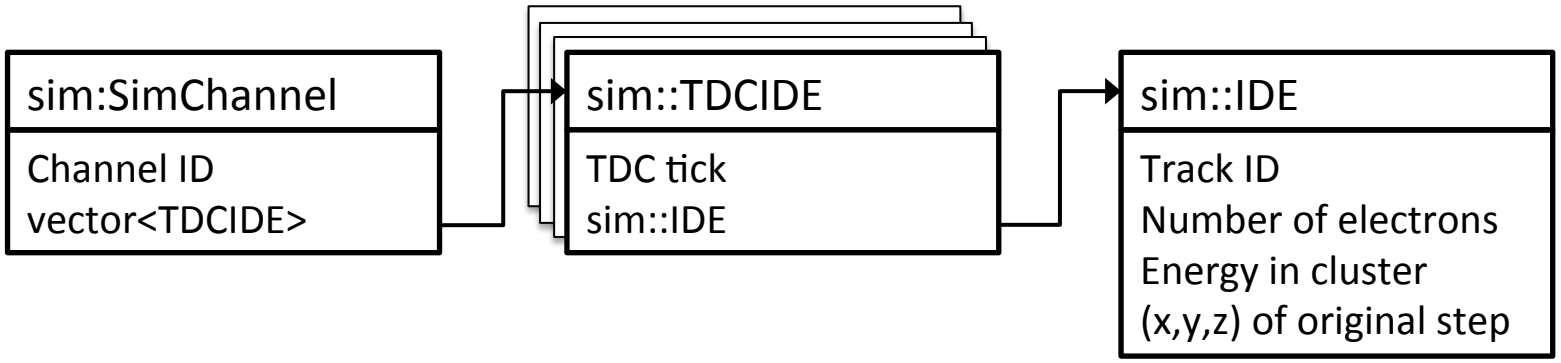


Accumulating electron clusters

The story so far:
Ionization -> number of electrons
Scintillation -> number of optical photons

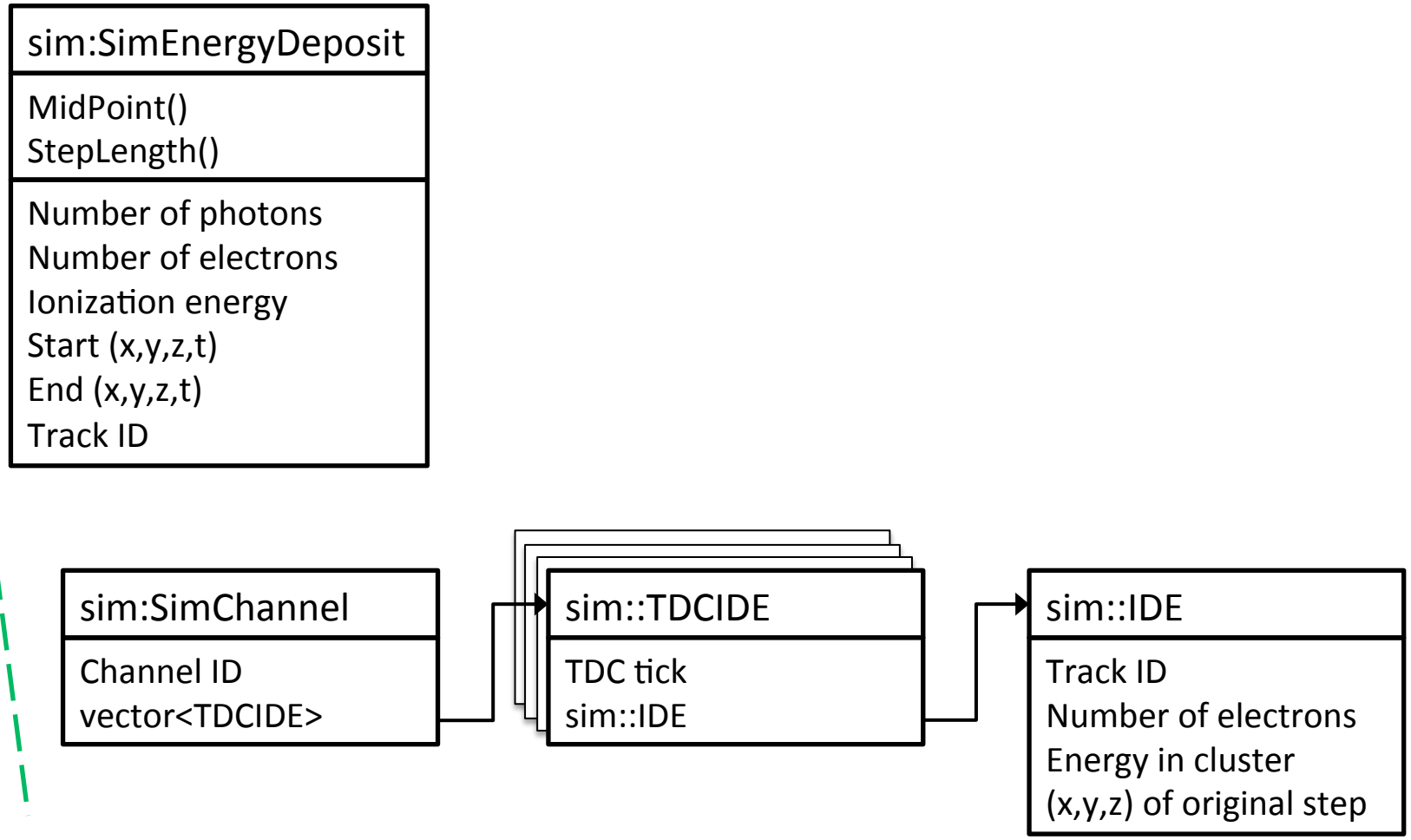
Number of electrons:
The electrons are grouped in clusters (20 electrons/cluster for uB).
Each cluster is individually drifted to a wire.
The space-charge effect is applied here.

The electron clusters are accumulated in the sim::SimChannel data product.



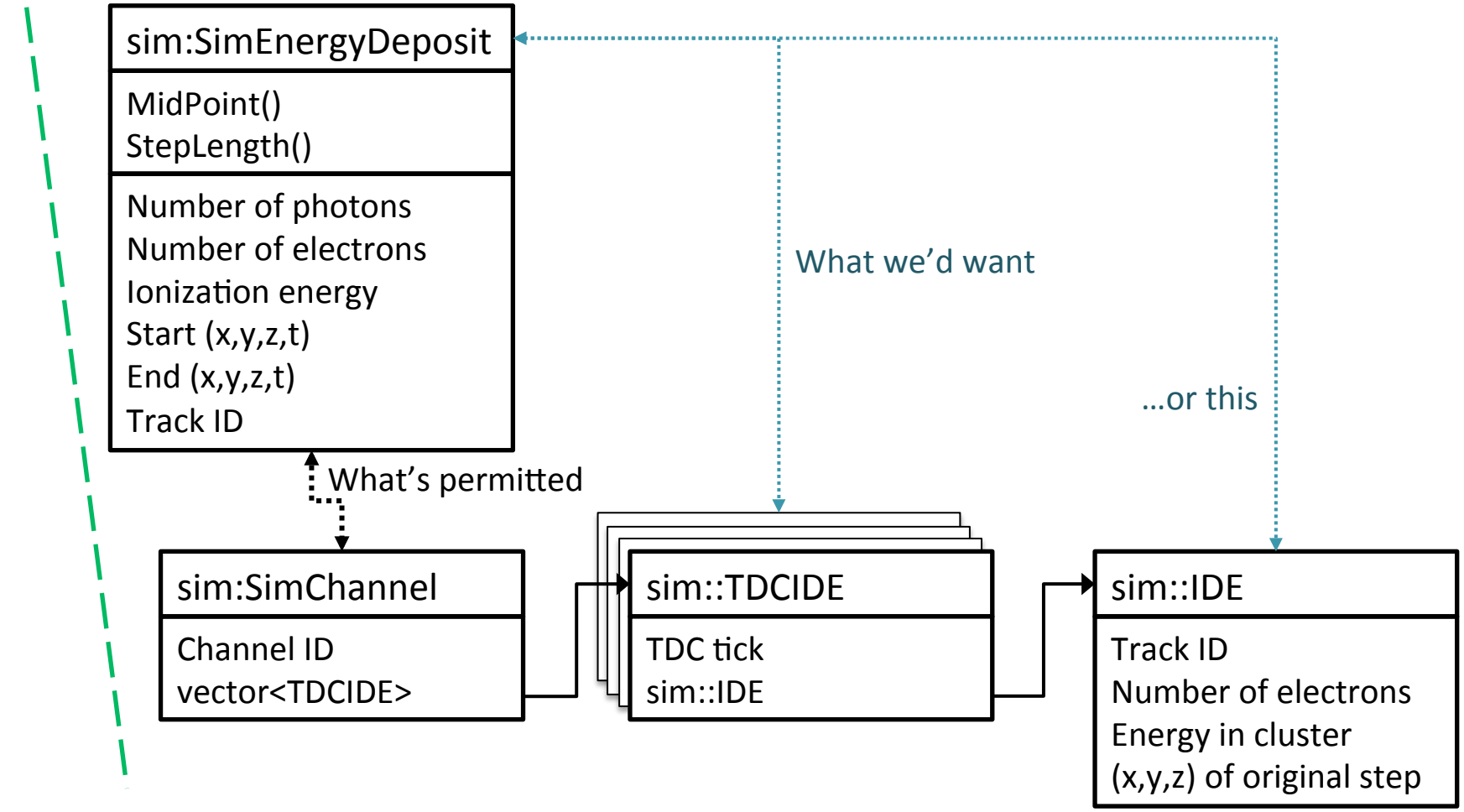
dE/dx

You can't get dE/dx from the contents of sim::SimChannel.
Proposal: Create a new per-step data product, sim::SimEnergyDeposit.



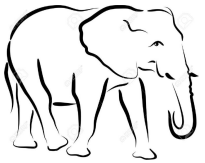
Assns

It seems natural to create associations between these two, but it's not clear if it will be useful.



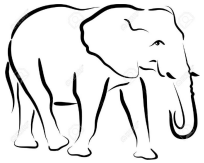
Assns

This would be a ”many-to-many” association.

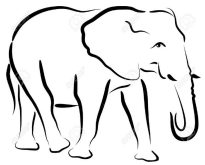


A single step can deposit electron clusters in more than one channel. Of course, a channel can contain contributions from multiple steps.

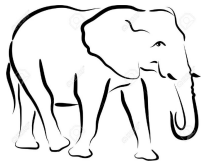
The perils of a new data product



The MergeSimSource module would have to be revised (by someone who can understand it; I can't).



The backtracking might have to be revised (if someone can figure out a reasonable scheme to go across the many-to-many associations).



MicroBooNE plans to strip SimChannels out of their simulations files when they're no longer needed for analysis. The same should probably be done with SimEnergyDeposit.

sim:SimEnergyDeposit
MidPoint() StepLength()
Number of photons Number of electrons Ionization energy Start (x,y,z,t) End (x,y,z,t) Track ID

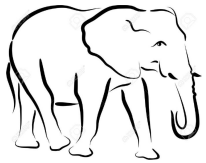
sim:SimChannel
Channel ID vector<TDCIDE>

sim::TDCIDE
TDC tick sim::IDE

sim::IDE
Track ID Number of electrons Energy in cluster (x,y,z) of original step

LArG4 RAM

Memory use is the biggest issue associated with running LArG4. How much impact will SimEnergyDeposit make on the LArG4 memory footprint?



Hard to say, because many of Hans' changes (e.g., eliminating parallels worlds and voxels) might reduce the RAM use.

My estimate: 14 words * 3000(?) steps = 42K/event. Is this right? We'll see…

sim:SimEnergyDeposit
MidPoint() StepLength()
Number of photons Number of electrons Ionization energy Start (x,y,z,t) End (x,y,z,t) Track ID

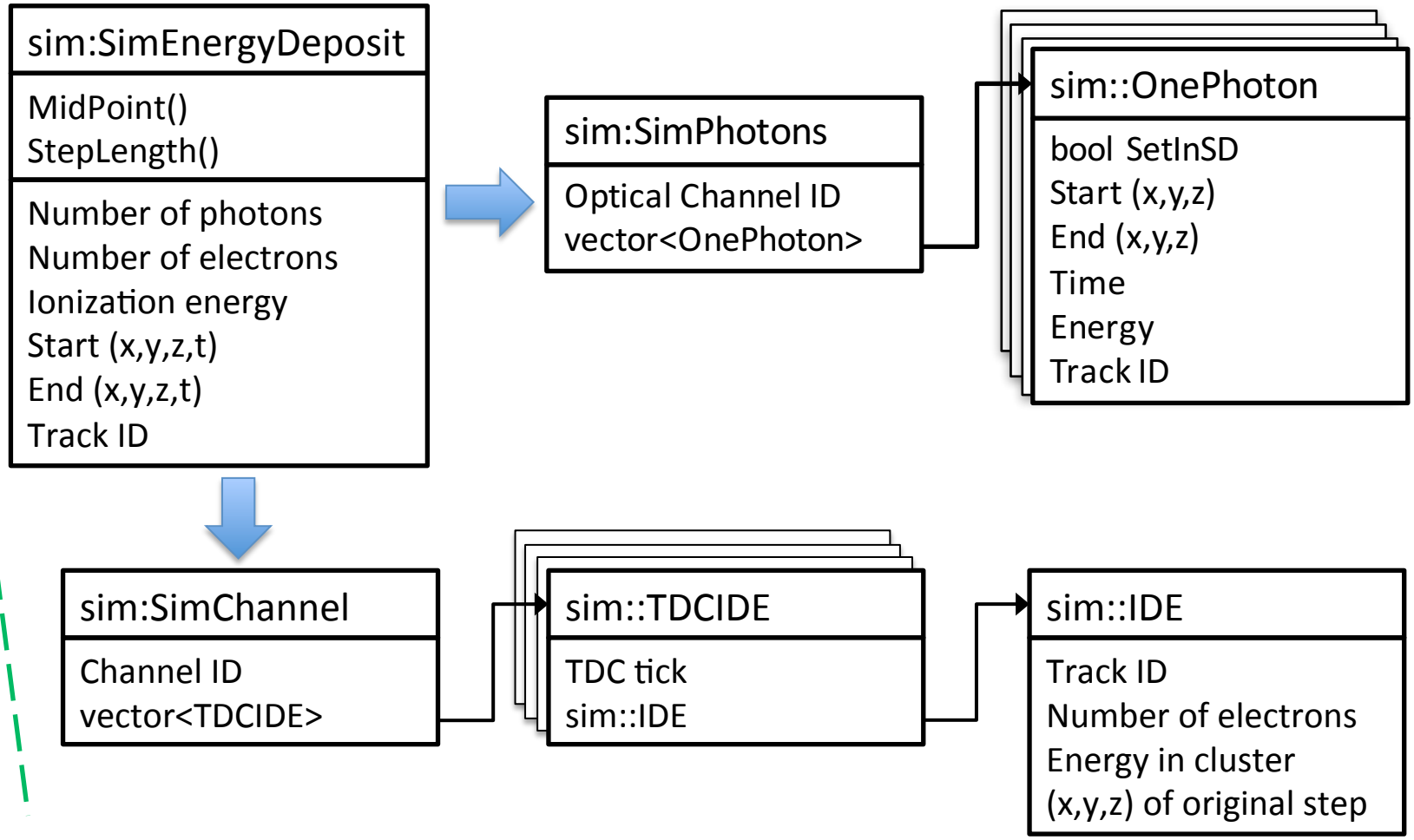
sim:SimChannel
Channel ID vector<TDCIDE>

sim::TDCIDE
TDC tick sim::IDE

sim::IDE
Track ID Number of electrons Energy in cluster (x,y,z) of original step

Present plan: new job modules

Have LArG4 just write out SimEnergyDeposit. Create the SimChannel and SimPhotons separately, each in their own module.



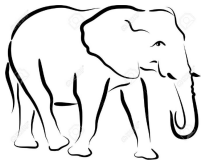
New job modules: Pros

Pros:

- LArG4 memory use will certainly go down; the anticipated size of the collection of SimEnergyDeposits is less than than that of SimChannels.
- The new modules (SimDriftElectrons and SimPhotonLookup) have the potential to be sped up by multiple threads.
- If an analysis does not use the photons, the SimPhotonLookup step can be omitted.
- They're convenient places to include post-Geant4 corrections.

New job steps: Cons

Cons:



- Adding new modules means that many modules downstream have to be revised due to new labels:
  - The SimChannels and SimPhotons will no longer be created by step “largeant”. This would mean just a change to the .fcl file.
  - If a module reads both SimChannels and MCParticles (e.g., the BackTrackers), they must be revised to accept a new module label parameter, since those two data products were no longer created by the same step.

- What will the memory impact be of adding two new client modules that read SimChannels, which represent one of the largest chunks of memory use? What is the ART memory model? If step A creates a data product, and step B and C read it, does the data product reside in memory two or three times?

The impact of these changes may be great enough that the Pros are not worth it. Opinions solicited!

Status

Hans Wenzel has made the necessary changes to nutools for the physics lists. He has his arms deep inside LArG4, working on setting up a single sensitive detector to generate the sim::SimEnergyDeposit data product.

William Seligman has set up the SimEnergyDeposit data product and the SimDriftElectrons module. Everything compiles, but he can't test it until Hans has the new LArG4 running. I'll start on SimPhotonLookup once Hans identifies the corresponding code in the "old" LArG4 (the old optical photon code is messy, since it's blended with the custom physics list).

Please don't ask for time estimates. Part of the answer will depend on a discussion of "Pros" vs. "Cons".

Backup Slides

Hans Wenzel's own words

What I did lately

- moved all setup of physics lists into nutools. I make use of new geant 4 PhysicsListfactory and the G4Opticalphysics, G4StepLimiter, NeutronKiller physics constructors.
- (done! Robert Hatcher will finish this up, also a new version of geant4 will simplify the code by allowing to use the names of Physicsconstructors when building the physics list)

Fast/full optical:

- Right now I am removing the photon library lookup from the physics process, and relocate it into a sensitive detector.
- Design a single sensitive detector that will do both the optical and charge simulation, sensitive to charge, to allow for future correlation between the two (e.g. NEST).
- Apply this sensitive detector both outside and inside the TPC at the GDML level in the "with wires" geometry. In case the e-field is 0 (as in the outside of the TPC) there is no charge just scintillation.
- provide sensitive detector that fills SimEnergyDeposit data object (provided by Bill Seligman)

test the whole setup with the lariat setup.