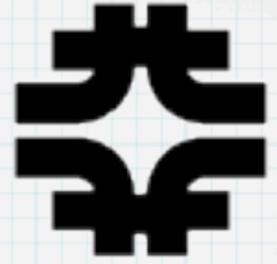




Geant 4



# NuMI Beam Simulation -- the reunification

---

Robert Hatcher  
Fermilab Computing Division

NuMI Beam Mtg 2012-05-03



# Tale of $N_{(N>2)}$ Cities

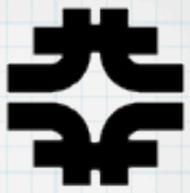


- Geant 4**
- The code has been “forked” a number of times
    - each time w/ a loss of prior CVS history
    - corrections to common code apply to each repository
  - Output Ntuple formats are inconsistent
    - trouble from the start: gnumi, g4numi, flugg
      - different names, types, capitalizations
      - variants of each of these make additional changes
    - core elements **are** the same
    - GENIE was coded to handle the original 3
      - Minerva hacked GENIE to handle their variant
      - near impossible to continue along same road in the future





# Codebase



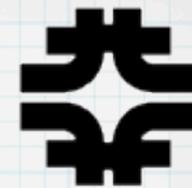
Geant 4



- Ideally, the code would get merged back into a single repository and experiments would use CVS (SVN, whatever) branches to handle alternatives
- if the objection is the repository location then we could start a new completely expt agnostic one
- would allow easier common fixes and re-integration
- BIG job to do it in the ideal manner
  - Robert just goes to his cubical and weeps every time he thinks about this task
- Put this on hold for now...



# Output Format



**Geant 4** ● This is the pressing issue ...

- GENIE needs this to handle new simulations
- experiments need some of the new features
- ...having a single format just makes life easier



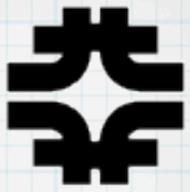
● Proposal [MINOS-DocDB-9070](#)

- currently still evolving - example code in document!
- general agreement on principles
- a tree of “dk2nu” class objects
  - represent hadron/muon that decays to a neutrino
  - class is roughly a union of all existing elements
- a tree of “dkmeta” class objects for metadata
  - name choice? tree or just one object in the file?





# General Principals



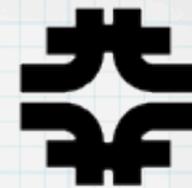
## Geant 4 ● Naming



- new name for tree `dk2nu` ( `dkmeta` )
- standardized variable names all lower case
- no truncation of variable names: `ndxdznear`, not `NdxdzNea`
- Everyone uses a versioned class header
  - additional info can be stored in parallel branches on the tree, but otherwise no unagreed upon changes
  - variables are ints, doubles, strings, vectors
    - no fixed sized arrays
  - class supplies a `Clear()` method to assign defaults



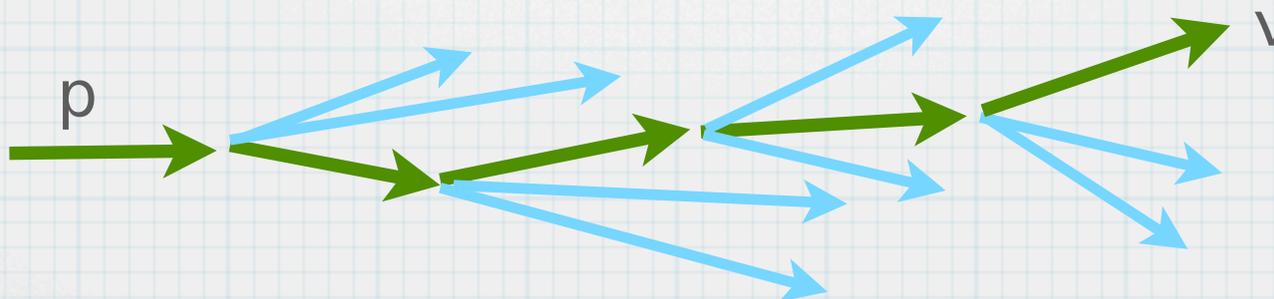
# General Principals



## Geant 4 • Biggest addition: ancestor chain

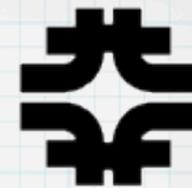


- concept from Minerva g4numi variant (w/ mods)
- one entry per state between proton and neutrino
  - $[0]$  = the initial proton;  $[n-1]$  = the final neutrino
- record
  - PDG code
  - starting & final momentum ...
  - starting position
    - don't need final position;  $\text{stop}[j] = \text{start}[j+1]$
  - other info up for debate





# Moving Forward



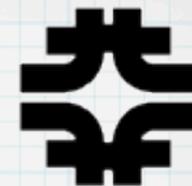
**Geant 4**

- Move simulations adiabatically
- gnumi (GEANT3) dead; no new generation?
  - uses hbook/zebra ntuples - hard to convert
  - GENIE will retain ability to read 3 legacy formats
- g4numi & flugg
  - relatively easy to adjust to write new format
    - variable names mostly
  - initially leave new (for that type) variables unfilled
  - metadata filling is the most new code
- flugg: rework code
  - convert text-to-ntuple script to compiled ACLiC code
    - straightforward for the most part
  - priority: add filling ancestor chain
    - necessary hadron production reweighting





# Use In GENIE



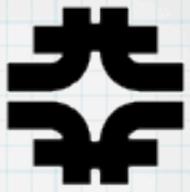
**Geant 4** ● GNuMIFlux  $\Rightarrow$  GDk2NuFlux

- no longer NuMI specific: LBNE, hopefully Booster too
- dk2nu class constitutes “passthrough” info
  - store this in expt’s data structures “as is” if desired
    - need to explore potential ROOT dictionary clash
      - copy in GENIE & expt framework (each can be used independently)
  - use of STL vector makes this better than fixed arrays because of copy mechanism (no explicit loop written by user)





# Timeline



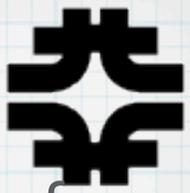
**Geant 4** ● Hopefully soon!

- mid-summer?





# Beam Simulation



## Geant 4



- Common Ntuple format
  - gnumi (geant3, obsolete)
  - flugg (g4+fluka, incomplete)
  - g4numi (+ minerva + lbne variants)
- Shared location for non-expt specific files
- Common mechanism for converting to GSimpleNtpFlux format
  - samples weighted files into form with unweighted rays
  - factorize computation speeds up actual generation
- Merge codebase back to one repository
  - snapshots taken at various times means history was lost, but fixes need to get propagated to multiple repositories
  - experiment based branches from common code allows desired flexibility
- Re-work flugg handling of alternatives
  - use run-time switches, not code recompilation
- Evolve flugg for full ancestor list
  - currently doesn't have all particles between initial proton and particle that decayed to give the neutrino
  - can't apply NA59/NA61 weights
- Physics choices
  - Geant4 PhysicsList alternatives
  - flugg fluka version (2011)
- Incorporation of external knowledge
  - NA49/NA61
  - cross expt hadron re-weighting
    - SKZP works for MINOS but not NOvA
  - revisit muon/hadron monitors?