UNIVERSAL NEUTRINO GENERATOR & GLOBAL FIT

Input from Generators - GENIE

Costas Andreopoulos

Generator Tools Workshop - Fermilab - Jan 8th, 2020

UNIVERSITY OF LIVERPOOL

Science & Technology Facilities Council

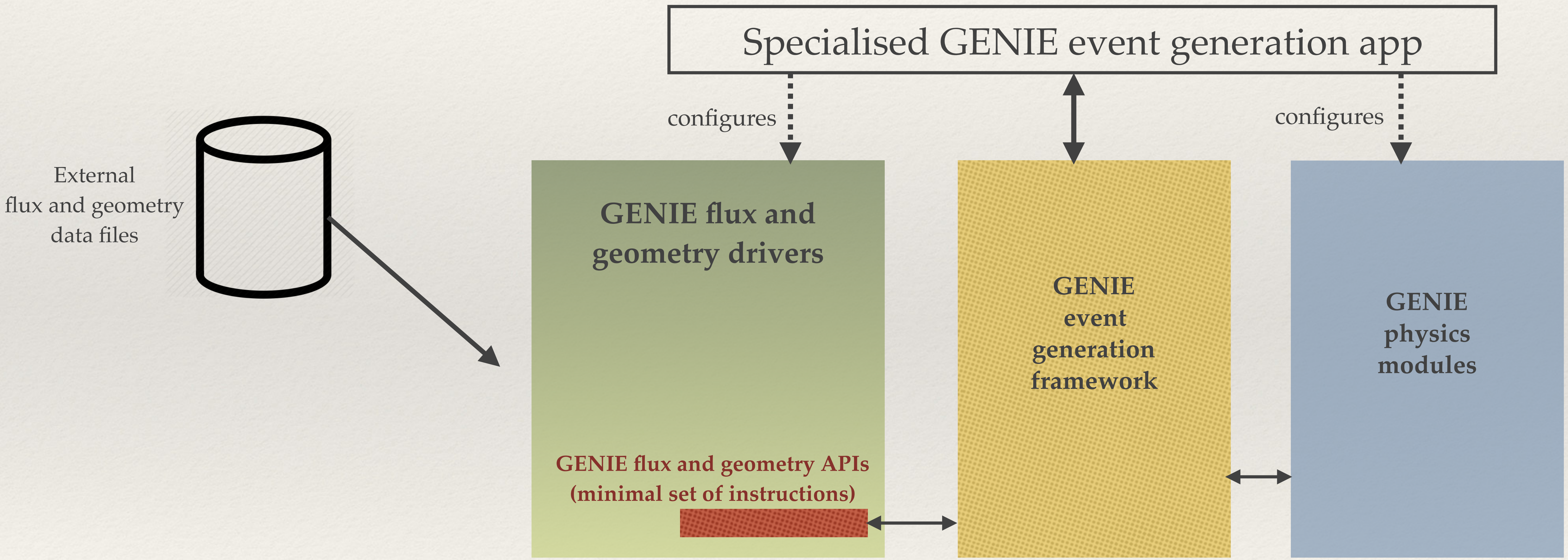
UK Research and Innovation

Questions by workshop organisers

- Give an overview of your flux/geometry driver functionality and input formats
- Are there any structural problems that would be needed to solved to use a common flux/geometry driver
- Review the output formats of your generator
- Are ISI and FSI done at separate stages in your generator.
  - If not, why not?
  - How difficult, mechanically, would this factorization be?

- Give an overview of your flux/geometry driver functionality and input formats

General scheme



GENIE flux and geom APIs

```
//
// define the GFluxI interface:
//
virtual const PDGCodeList &    FluxParticles (void) = 0; ///< declare list of flux neutrinos that can be generated (for init.
virtual double                 MaxEnergy     (void) = 0; ///< declare the max flux neutrino energy that can be generated (for
virtual bool                   GenerateNext  (void) = 0; ///< generate the next flux neutrino (return false in err)
virtual int                    PdgCode       (void) = 0; ///< returns the flux neutrino pdg code
virtual double                 Weight        (void) = 0; ///< returns the flux neutrino weight (if any)
virtual const TLorentzVector & Momentum      (void) = 0; ///< returns the flux neutrino 4-momentum
virtual const TLorentzVector & Position      (void) = 0; ///< returns the flux neutrino 4-position (note: expect SI rather th
virtual bool                   End           (void) = 0; ///< true if no more flux nu's can be thrown (eg reaching end of bea
virtual long int               Index         (void) = 0; ///< returns corresponding index for current flux neutrino (e.g. for
virtual void                   Clear            (Option_t * opt   ) = 0; ///< reset state variables based on opt
virtual void                   GenerateWeighted (bool gen_weighted) = 0; ///< set whether to generate weighted or unweighted
```

```
// define the GeomAnalyzerI interface

virtual const PDGCodeList &
          ListOfTargetNuclei (void) = 0;

virtual const PathLengthList &
          ComputeMaxPathLengths (void) = 0;
virtual const PathLengthList &
          ComputePathLengths (
            const TLorentzVector & x, const TLorentzVector & p) = 0;
virtual const TVector3 &
          GenerateVertex (
            const TLorentzVector & x, const TLorentzVector & p, int tgtpdg) = 0;
```

GENIE flux drivers

- **A number of flux drivers are included in public releases**
  - Barr - Gaisser - Lipari - Robbins - Stanev (**BGLRS**) atm. flux driver
  - Ferrari - Sala - Battistoni - Montaruli (**FLUKA**) atm. flux driver
  - Honda - Sajjad Athar - Kajita - Kasahara - Midorikawa (**HAKKM**) atm. flux driver
  - Driver for integration into GENIE of the outputs of a detailed J-PARC neutrino beam-line simulation [**JNUBEAM**].
  - Driver for integration into GENIE of the outputs of a detailed NuMI neutrino beam-line simulation [**GNuMI**].
  - **A generalised driver** for a simple, "cylindrical", multi-neutrino flavour flux where the direction, size and lateral profile are customisable and the energy distribution of each flavour is described by a **1-D histogram**
  - **A generalised driver** for a multi-neutrino flavour **flux represented as an n-tuple** (a series of *neutrino flux rays*) that can be used when there are substantial correlations between the neutrino position, energy, direction
- **Severals other tools in private versions and / or in user codes**

GENIE geometry drivers

- **Two geometry drivers are included in public releases**
  - ROOT geometry driver - the main workhorse
    - TGeoManager
    - GEANT geometries can be exported in this format
  - "Point geometry" driver
    - Allows handling simple target mixtures (list of targets each with its corresponding weight fraction) with no spatial information in the exact same framework

Less diversity - but if there was any other different geometry description used by the community we would support it with a dedicated geometry driver

Building on top of the flux and geometry APIs

- The flux and geometry APIs provide a protocol for obtaining the information required **for the purpose of event generation**

The fundamental problem here is to compute a complex multi-dimensional integral

$$N_{ev} \propto \int dE_\nu dcos\theta_\nu d\phi_\nu dxdydt \sum_{f(lavor)} \frac{d^6\Phi_f(E_\nu, cos\theta_\nu, \phi_\nu, x, y, t)}{dE_\nu dcos\theta_\nu d\phi_\nu dxdydt} \int_0^\infty ds \sum_{i(sotope)} \frac{\rho(\vec{r}) w_i(\vec{r}) \sigma^{tot}_{f;i}(E_\nu)}{A_i(\vec{r})}$$

for which an analytical answer is not possible.

Flux description   Detector description

- That protocol specifies a **very minimal set of operations** - All we need is:
  - **Throw "flux rays"** (flavour, momentum and position 4-vectors) from input p.d.f.s
  - Navigate through the geometry and **compute density-weighted path-lengths** for each isotope
- A **much larger set of operations** is required for initialising, configuring , manipulating different types of fluxes - not part of the API: Specific to different types of fluxes and exploited by different types of event generation apps, specialised for different tasks.

Using flux and geo drivers in a specialised event generation app

- A **much larger set of operations** is required for initialising, configuring , manipulating different types of fluxes - not part of the API
- It is worth considering briefly a couple of examples

```
FLUKA:/some/path/sdave_numu07.dat[14],/some/path/sdave_nue07.dat[12],…
```

```
/some/path/duneFD.root
```

or

```
1000080160[0.8879],1000010010[0.1121]
```

Synopsis

```
$ gevgen_atmo
    -f flux -g geometry
    [-R rotation_from_topocentric_hz_frame]
    [-t geometry_top_volume_name] [-m max_path_lengths_xml_file]
    [-L geometry_length_units] [-D geometry_density_units]
    <-n number_of_events, -e exposure_in_terms_of_kton_x_yrs>
    [-E energy_range] [-o output_event_file_prefix] [-r run#]
    [-seed random_number_seed] [--cross-sections xml_file] [--event-generator-list list_name]
    [--message-thresholds xml_file] [--unphysical-event-mask mask] [--event-record-print-level level]
    [--mc-job-status-refresh-rate rate] [--cache-file root_file]
    [-h]
```

Using flux and geo drivers in a specialised event generation app

- A **much larger set of operations** is required for initialising, configuring , manipulating different types of fluxes - not part of the API
- It is worth considering briefly a couple of examples

Enable / disable volumes, eg "-Cryostat -Rock"

plus, ability to define and generate events in arbitrary fiducial volumes that do not correspond to any actual geometry volume

Rotation: Topocentric horizontal -> user-defined coordinate system

The Euler angles are input as a comma separated list. The general syntax for specifying the rotation is: '`-R convention:phi,theta,psi`' where '`convention`' is either X (for X-convention), Y (for Y-convention), X^-1 or Y^-1 (as previously, but using the inverse rotation matrix instead).

Example 1:
To set the Euler angles φ=3.14, ϑ=1.28, ψ=1.0 using the X-convention, type: '`-R 3.14,1.28,1.0`', or '`-R X:3.14,1.28,1.0`'.

Example 2:
To set the Euler angles φ=3.14, ϑ=1.28, ψ=1.0 using the Y-convention, type: '`-R Y:3.14,1.28,1.0`'.

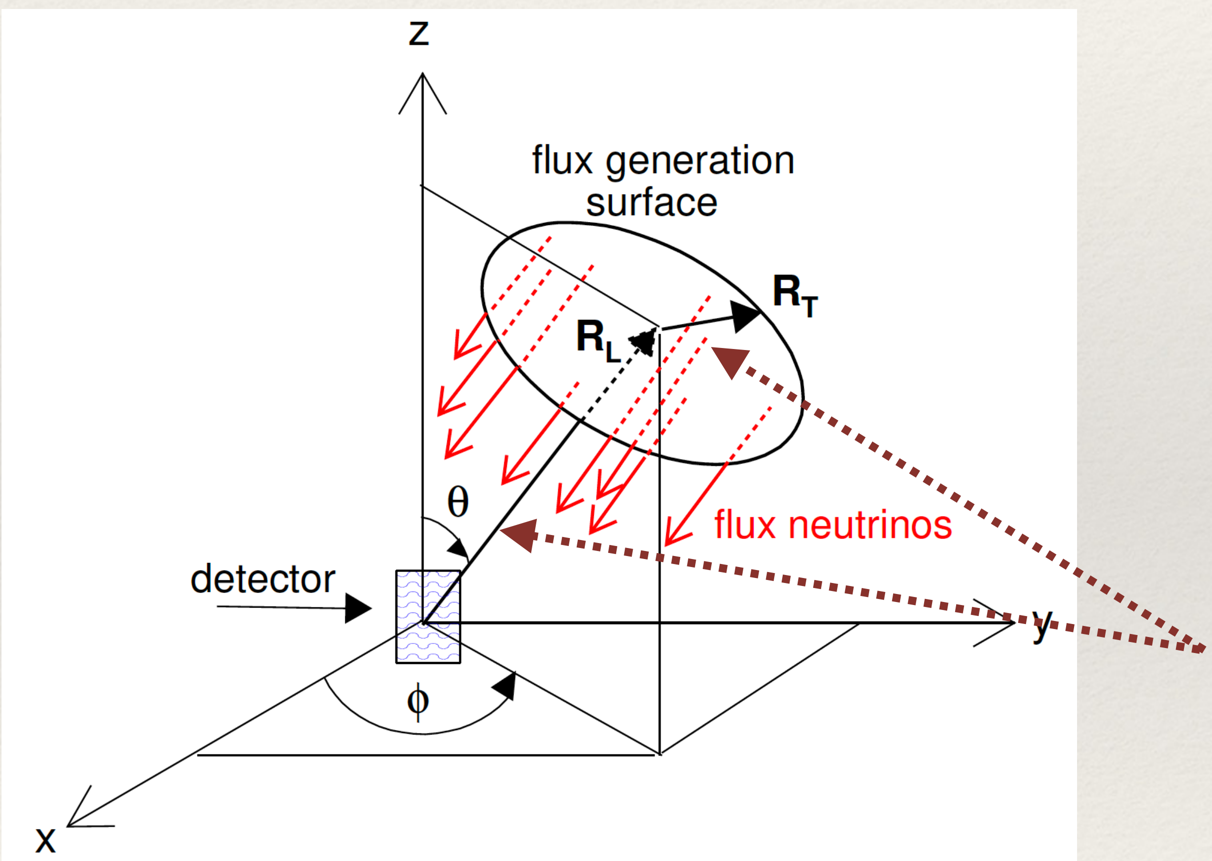
Example 3:
To set the Euler angles φ=3.14, ϑ=1.28, ψ=1.0 using the Y-convention, and then use the inverse rotation matrix, type: '`-R Y^-1:3.14,1.28,1.0`'.

Synopsis

```
$ gevgen_atmo
    -f flux -g geometry
    [-R rotation_from_topocentric_hz_frame]
    [-t geometry_top_volume_name] [-m max_path_lengths_xml_file]
    [-L geometry_length_units] [-D geometry_density_units]
    <-n number_of_events, -e exposure_in_terms_of_kton_x_yrs>
    [-E energy_range] [-o output_event_file_prefix] [-r run#]
    [-seed random_number_seed] [--cross-sections xml_file] [--event-generator-list list_name]
    [--message-thresholds xml_file] [--unphysical-event-mask mask] [--event-record-print-level level]
    [--mc-job-status-refresh-rate rate] [--cache-file root_file]
    [-h]
```

Using flux and geo drivers in a specialised event generation app

- A **much larger set of operations** is required for initialising, configuring , manipulating different types of fluxes - not part of the API
- It is worth considering briefly a couple of examples



- Definition of *flux ray generation surface* for atmospheric neutrinos

Customisable radii

- Similarly, **<u>all flux driver</u>s have customisation options, or use conventions, specific to the given type of flux**

- Are there any structural problems that would be needed to solved to use a common flux/geometry driver?

- If only I was making choices on the amount of information provided here …
- There is no concrete proposal / specification of how that system would look like
    - Can not comment on whether there would be technical limitations
    - For instance, if the new system was art-based (possible, given the strong FNAL focus of this workshop), we wouldn't touch it - No need for us to impose new requirements to our many non-FNAL user communities.

- Even if there was no technical limitation, it is very hard to see why we would want to use an *unknown new system* to replace of our **existing**, **native**, **proven**, **well-developed** and **well-known** tools that **already underpin the GENIE interfaces to all experiments!!**

- Almost certainly, **options #2a and #2b** (outlined in my previous talk) define the **context in which we could engage in joint developments**

- Review the output formats of your generator

Output formats

The native GENIE event format is called **GHEP -** a suitably customised *STDHEP*-like format

A **GHepRecord** is a TClonesArray of **GHepParticles**

- It is a special kind of TClonesArray, where **entries are automatically shuffled** to make sure the daughter list of any particle in the event occupies a consecutive set of slots.
- In addition to lists of particles it **also includes information with event-wide scope**
  - Event vertex in detector coordinate system, event weight, error flags, corresponding x-section
- It also includes **redundant information in a hierarchical set of objects** (more later)

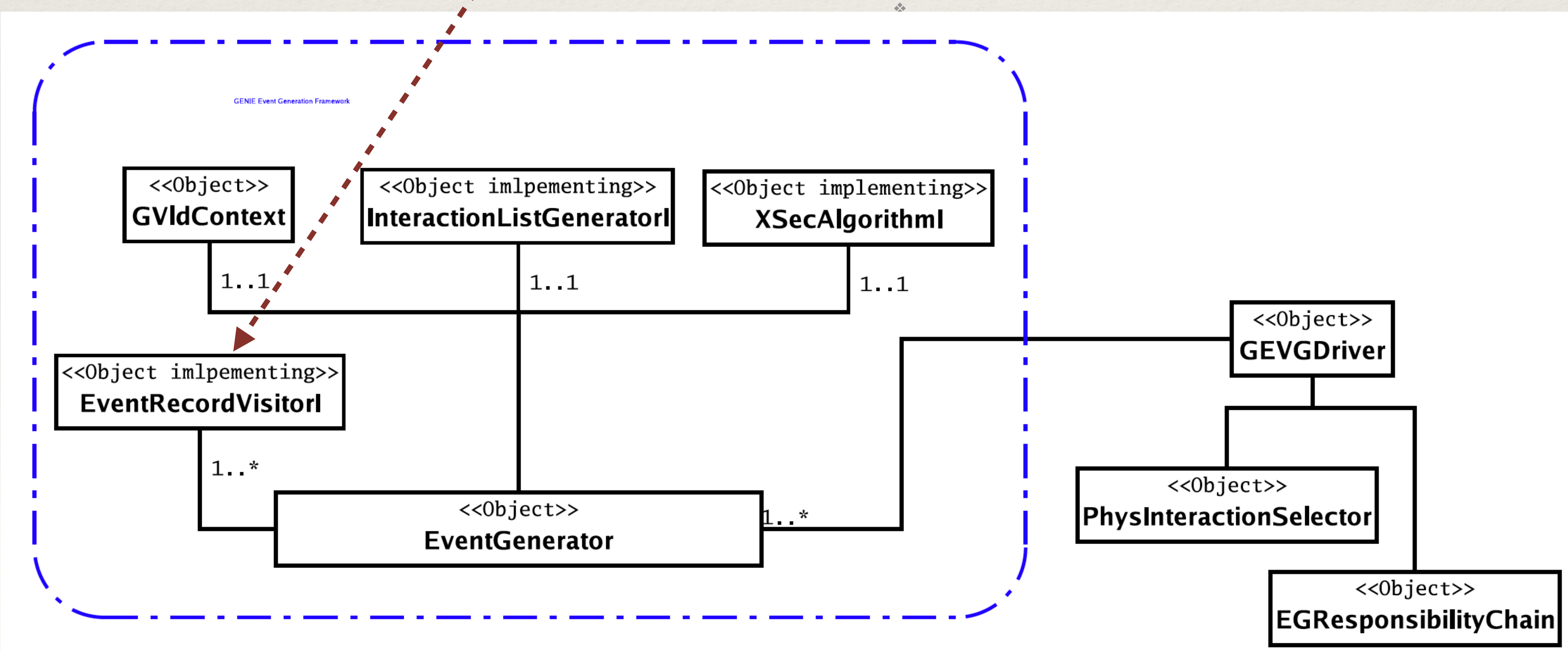
GHEP-specifics deeply ingrained in all GENIE generation codes

```
class EventRecord : public GHepRecord {

  ...
  ...

  void AcceptVisitor (EventRecordVisitorI * visitor);
```

- Each event generator includes a list of event generation modules
- Each module "*visits*" and operates on the current event (Visitor design pattern)
- **Achieves a uniform interface for a very diverse set of physics modules** (kinematic selection, FSI, hadronization, particle decays, Fermi motion, Pauli blocking, …)
-



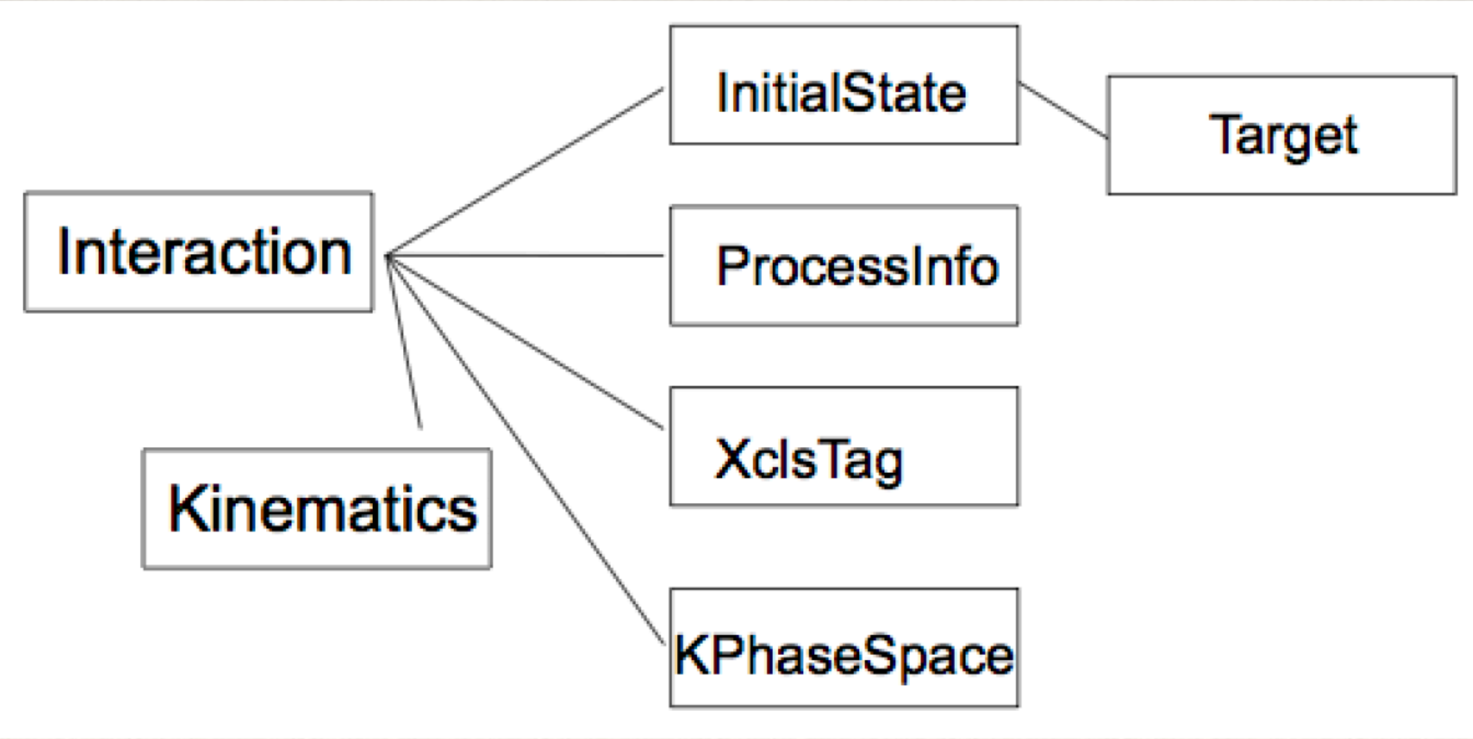
Redundant information - Interaction

- The **GHEP** event record **specifies all information about an event**
- It is the main input to **and** output of GENIE MC event generators / event generation modules
- As part of MC event generation we also do **calculations** (e.g. cross-section or form factor calculations)
- For these calculations there are **several use cases that do not involve MC generation**
  - Typically, they require a very small subset of the information stored in the event
  - We need to be able to run these calculations even when we don't have a full event

This info is stored in a hierarchical set of objects (**Interaction** and other objects there in)

Whereas a GHEP event object is passed to all GENIE MC generation modules, **an Interaction object is passed to nearly all GENIE calculations**

An Interaction object is **attached to each event**



Redundant information - Interaction

So, for example, if a QE generator wants to take the currently configured QE x-section model and evaluate a x-section value, it does something like

```
Interaction * ccqe = Interaction::QELCC(kPdgTgtO16,kPdgNeutron,kPdgNuMu); // numu + (n)C12
ccqe->InitStatePtr()->SetProbeE(E); // in LAB-frame (E,0,0,E)
ccqe->KinePtr()->SetQ2(Q2);


RunningThreadInfo * reinfo = RunningThreadInfo::Instance();
const EventGeneratorI * evg = rtinfo->RunningThread();
XSecAlgorithmI * xsec_model = evg->CrossSectionAlg();

// print dsig/dQ2
cout << xsec_model->XSec(interaction, kPSQ2fE) / (1E-38 * units:cm2) << endl;
```

This is event record feature is deeply ingrained in GENIE code

Care required to make sure the GHEP record and the attached Interaction remain in sync

Benefits (simple interface to most calculations, decoupling from event generation) is worth the trouble

GHEP event record - example 1

Particles include more info than shown here

```
|-----------------------------------------------------------------------------------------------------------------------|
|GENIE GHEP Event Record [print level:   3]                                                                           |
|-----------------------------------------------------------------------------------------------------------------------|
| Idx |          Name | Ist |        PDG |   Mother  | Daughter  |      Px |      Py |      Pz |       E |      m  |
|-----------------------------------------------------------------------------------------------------------------------|
|   0 |         nu_mu |   0 |         14 |  -1 |  -1 |   4 |   4 |   0.000 |   0.000 | 100.000 | 100.000 |   0.000 |
|   1 |           C12 |   0 | 1000060120 |  -1 |  -1 |   2 |   3 |   0.000 |   0.000 |   0.000 |  11.175 |  11.175 |
|   2 |        proton |  11 |       2212 |   1 |  -1 |   5 |   5 |   0.132 |  -0.033 |   0.151 |   0.920 | **0.938 | M = 0.898
|   3 |           B11 |   2 | 1000050110 |   1 |  -1 |  26 |  26 |  -0.132 |   0.033 |  -0.151 |  10.255 |  10.253 |
|   4 |         nu_mu |   1 |         14 |   0 |  -1 |  -1 |  -1 |  -1.243 |   0.464 |  39.229 |  39.251 |   0.000 | P = (0.032,-0.012,-0.999
|   5 |      HadrSyst |  12 | 2000000001 |   2 |  -1 |   6 |   7 |   1.376 |  -0.497 |  60.922 |  61.669 | **0.000 | M = 9.457
|   6 |             d |  12 |          1 |   5 |  -1 |   8 |  16 |   1.375 |  -0.497 |  60.878 |  60.897 |   0.330 |
|   7 |          uu_1 |  12 |       2203 |   5 |  -1 |  -1 |  -1 |   0.001 |  -0.000 |   0.044 |   0.773 |   0.771 |
|   8 |       neutron |  14 |       2112 |   6 |   7 |  17 |  17 |   0.711 |  -0.026 |  37.725 |  37.744 |   0.940 | FSI = 1
|   9 |           pi- |  14 |       -211 |   6 |   7 |  18 |  18 |   0.178 |  -0.221 |   2.294 |   2.316 |   0.140 | FSI = 1
|  10 |            K+ |  14 |        321 |   6 |   7 |  19 |  19 |  -0.230 |  -0.230 |   0.300 |   0.663 |   0.494 | FSI = 1
|  11 |           pi+ |  14 |        211 |   6 |   7 |  20 |  20 |   0.705 |  -0.087 |   2.248 |   2.362 |   0.140 | FSI = 1
|  12 |           pi0 |  14 |        111 |   6 |   7 |  21 |  21 |  -0.148 |  -0.297 |   1.878 |   1.912 |   0.135 | FSI = 1
|  13 |   antineutron |  14 |      -2112 |   6 |   7 |  22 |  22 |   0.119 |  -0.304 |   7.275 |   7.343 |   0.940 |
|  14 |           pi- |  14 |       -211 |   6 |   7 |  23 |  23 |  -0.058 |   0.199 |   1.970 |   1.986 |   0.140 | FSI = 1
|  15 |        proton |  14 |       2212 |   6 |   7 |  24 |  24 |   0.108 |   0.442 |   4.954 |   5.063 |   0.938 | FSI = 1
|  16 |           pi0 |  14 |        111 |   6 |   7 |  25 |  25 |  -0.009 |   0.026 |   2.277 |   2.281 |   0.135 | FSI = 1
|  17 |       neutron |   1 |       2112 |   8 |   7 |  -1 |  -1 |   0.711 |  -0.026 |  37.725 |  37.744 |   0.940 |
|  18 |           pi- |   1 |       -211 |   9 |   7 |  -1 |  -1 |   0.178 |  -0.221 |   2.294 |   2.316 |   0.140 |
|  19 |            K+ |   1 |        321 |  10 |   7 |  -1 |  -1 |  -0.230 |  -0.230 |   0.300 |   0.663 |   0.494 |
|  20 |           pi+ |   1 |        211 |  11 |   7 |  -1 |  -1 |   0.705 |  -0.087 |   2.248 |   2.362 |   0.140 |
|  21 |           pi0 |   1 |        111 |  12 |   7 |  -1 |  -1 |  -0.148 |  -0.297 |   1.878 |   1.912 |   0.135 |
|  22 |   antineutron |   1 |      -2112 |  13 |   7 |  -1 |  -1 |   0.119 |  -0.304 |   7.275 |   7.343 |   0.940 |
|  23 |           pi- |   1 |       -211 |  14 |   7 |  -1 |  -1 |  -0.058 |   0.199 |   1.970 |   1.986 |   0.140 |
|  24 |        proton |   1 |       2212 |  15 |   7 |  -1 |  -1 |   0.108 |   0.442 |   4.954 |   5.063 |   0.938 |
|  25 |           pi0 |   1 |        111 |  16 |   7 |  -1 |  -1 |  -0.009 |   0.026 |   2.277 |   2.281 |   0.135 |
|  26 |      HadrBlob |  15 | 2000000002 |   3 |  -1 |  -1 |  -1 |  -0.132 |   0.033 |  -0.151 |  10.255 | **0.000 | M = 10.253
|-----------------------------------------------------------------------------------------------------------------------|
|       Fin-Init:                                                 |  -0.000 |   0.000 |  -0.000 |  -0.000 |           |
|-----------------------------------------------------------------------------------------------------------------------|
|       Vertex:          nu_mu @ (x =     0.00000 m, y =     0.00000 m, z =     0.00000 m, t =    0.000000e+00 s) |
|-----------------------------------------------------------------------------------------------------------------------|
| Err flag [bits:15->0] : 0000000000000000    |  1st set:                                                     none |
| Err mask [bits:15->0] : 1111111111111111    |  Is unphysical:    NO |   Accepted:   YES                          |
|-----------------------------------------------------------------------------------------------------------------------|
| sig(Ev) =       3.49185e-37 cm^2  | d2sig(x,y;E)/dxdy =      5.48927e-37 cm^2      | Weight =             1.00000 |
|-----------------------------------------------------------------------------------------------------------------------|
```

GHEP event record - example 1

Printout of associated / attached **Interaction** summary of previous event

```
--------------------------------------------------------------------------------------------
GENIE Interaction Summary
--------------------------------------------------------------------------------------------
[-] [Init-State]
 |--> probe        : PDG-code = 14 (nu_mu)
 |--> nucl. target : Z = 6, A = 12, PDG-Code = 1000060120 (C12)
 |--> hit nucleon  : PDC-Code = 2212 (proton)
 |--> hit quark    : PDC-Code = 1 (d) [valence]
 |--> probe 4P     : (E =   100.000000, Px =     0.000000, Py =     0.000000, Pz =   100.000000)
 |--> target 4P    : (E =    11.174863, Px =     0.000000, Py =     0.000000, Pz =     0.000000)
 |--> nucleon 4P   : (E =     0.920295, Px =     0.132484, Py =    -0.033330, Pz =     0.150874)
[-] [Process-Info]
 |--> Interaction : Weak[NC]
 |--> Scattering  : DIS
[-] [Kinematics]
 |-->  *Running* Hadronic invariant mass W = 9.456865
 |--> *Selected* Bjorken x = 0.048209
 |--> *Selected* Inelasticity y = 0.605103
 |--> *Selected* Momentum transfer Q2 (>0) = 4.489036
 |--> *Selected* Hadronic invariant mass W = 9.456865
[-] [Exclusive Process Info]
 |--> charm prod.  : false |--> strange prod.  : false
 |--> f/s nucleons : N(p) = 0 N(n) = 0
 |--> f/s pions    : N(pi^0) = 0 N(pi^+) = 0 N(pi^-) = 0
 |--> f/s Other    : N(gamma) = 0 N(Rho^0) = 0 N(Rho^+) = 0 N(Rho^-) = 0
 |--> resonance    : [not set]
--------------------------------------------------------------------------------------------
```

GHEP event record - example 2

A non-neutrino event

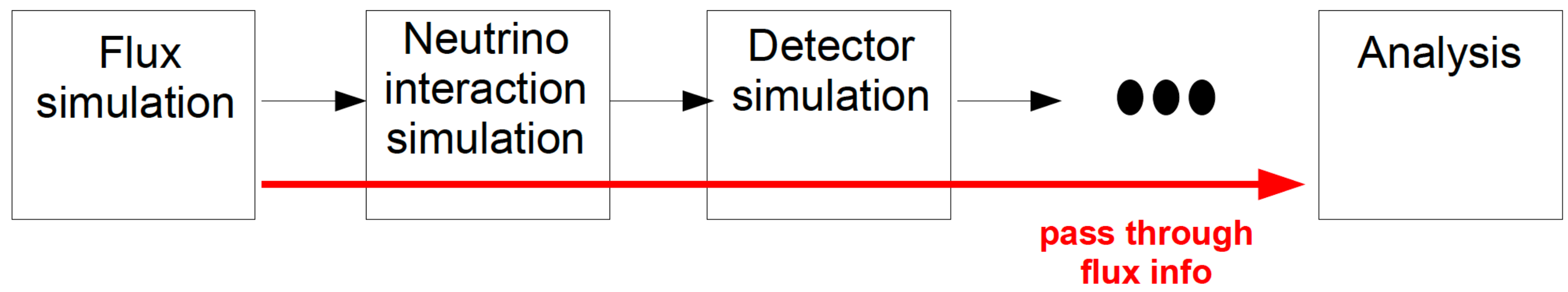
```
|--------------------------------------------------------------------------------------------------------------|
|GENIE GHEP Event Record [print level:   3]                                                                    |
|--------------------------------------------------------------------------------------------------------------|
| Idx |          Name | Ist |        PDG |   Mother  | Daughter  |       Px |       Py |       Pz |        E |        m |
|--------------------------------------------------------------------------------------------------------------|
|   0 |           O16 |   0 | 1000080160 |  -1 |  -1 |   1 |   2 |    0.000 |    0.000 |    0.000 |   14.895 |   14.895 |
|   1 |        proton |   3 |       2212 |   0 |  -1 |   3 |   5 |   -0.059 |    0.072 |   -0.093 |    0.926 | **0.938 | M = 0.916
|   2 |           N15 |   2 | 1000070150 |   0 |  -1 |   9 |   9 |    0.059 |   -0.072 |    0.093 |   13.970 |   13.969 |
|   3 |            e+ |   1 |        -11 |   1 |  -1 |  -1 |  -1 |   -0.291 |    0.191 |   -0.077 |    0.357 |    0.001 |
|   4 |           pi+ |  14 |        211 |   1 |  -1 |   6 |   7 |    0.316 |    0.029 |    0.014 |    0.347 |    0.140 | FSI = 2
|   5 |           pi- |  14 |       -211 |   1 |  -1 |   8 |   8 |   -0.084 |   -0.148 |   -0.029 |    0.222 |    0.140 | FSI = 1
|   6 |           pi0 |   1 |        111 |   4 |  -1 |  -1 |  -1 |   -0.127 |   -0.054 |   -0.166 |    0.254 |    0.135 |
|   7 |        proton |   1 |       2212 |   4 |  -1 |  -1 |  -1 |    0.405 |   -0.077 |    0.073 |    1.027 |    0.938 |
|   8 |           pi- |   1 |       -211 |   5 |  -1 |  -1 |  -1 |   -0.084 |   -0.148 |   -0.029 |    0.222 |    0.140 |
|   9 |      HadrBlob |  15 | 2000000002 |   2 |  -1 |  -1 |  -1 |    0.097 |    0.088 |    0.199 |   13.035 | **0.000 | M = 13.033
|--------------------------------------------------------------------------------------------------------------|
|       Fin-Init:                                                 |    0.000 |   -0.000 |    0.000 |    0.000 |          |
|--------------------------------------------------------------------------------------------------------------|
| Err flag [bits:15->0] : 0000000000000000     |  1st set:                                                none |
| Err mask [bits:15->0] : 1111111111111111     |  Is unphysical:    NO |   Accepted:   YES                      |
|--------------------------------------------------------------------------------------------------------------|
| sig(Ev) =       0.00000e+00 cm^2  | dsig(Ev;{K_s})/dK   =     0.00000e+00 cm^2/{K}  | Weight =         1.00000 |
|--------------------------------------------------------------------------------------------------------------|
```

Conversions to other formats

- **GENIE includes an app (gntpc) that can convert GHEP events to alternative formats:**
  - **GST:** flat n-tuple with tonnes of info for each event
  - **GXML:** A custom XML-based event format
  - **GHEP_MOCK_DATA:** GHEP formatted event where all info other than final state particles is stripped out
  - **NUANCE_TRACKER:** The original tracker format (used by NUANCE)
  - **T2K_TRACKER:** A variation of the tracker format with tweaks required for GENIE/SK interface
  - **ROOTRACKER:** A bare-ROOT STDHEP/GHEP-like event record
  - **ROOTRACKER_MOCK_DATA**
  - **T2K_ROOTRACKER:** A ROOTRACKER event format + JNUBEAM flux pass-through info
  - **NUMI_ROOTRACKER:** A ROOTRACKER event format + GNUMI flux pass-through info

  - **Any other can be added**

Flux pass-through information

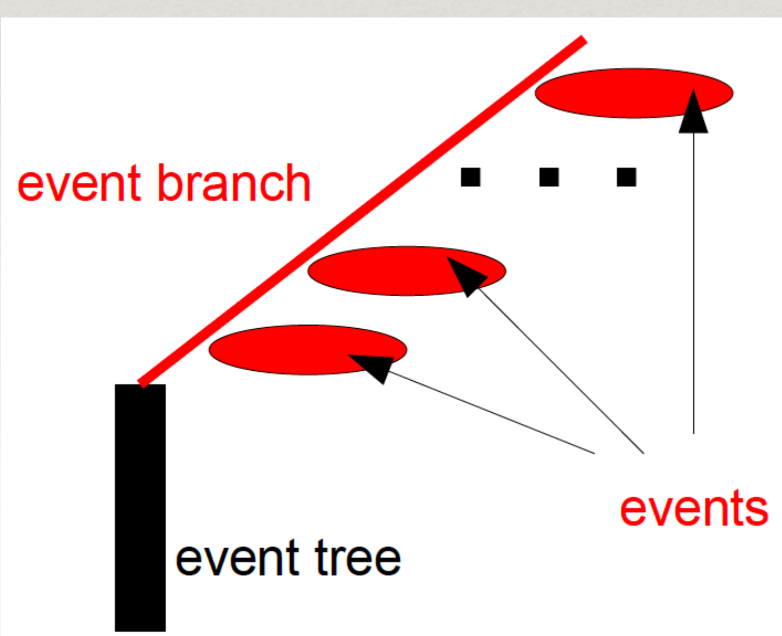


Neutrino generator sees flux simulation

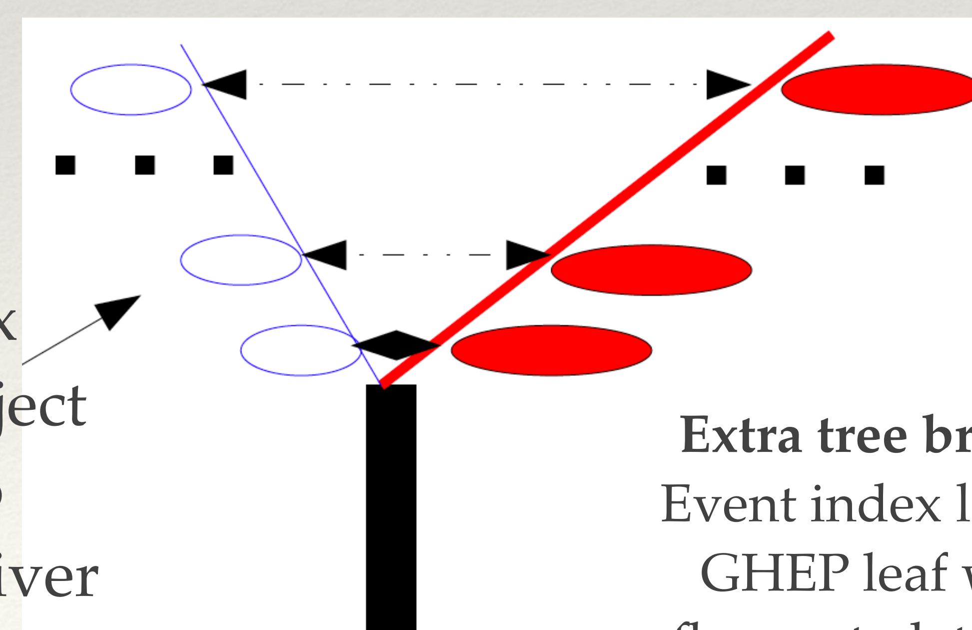
Neutrino generator has no use for certain flux information (eg, parent pion decay kinematics)

But the **neutrino generator** should **pass through all the flux info needed by analyses**

Output GENIE GHEP event tree
without flux metadata



Output GENIE GHEP event tree
With flux metadata

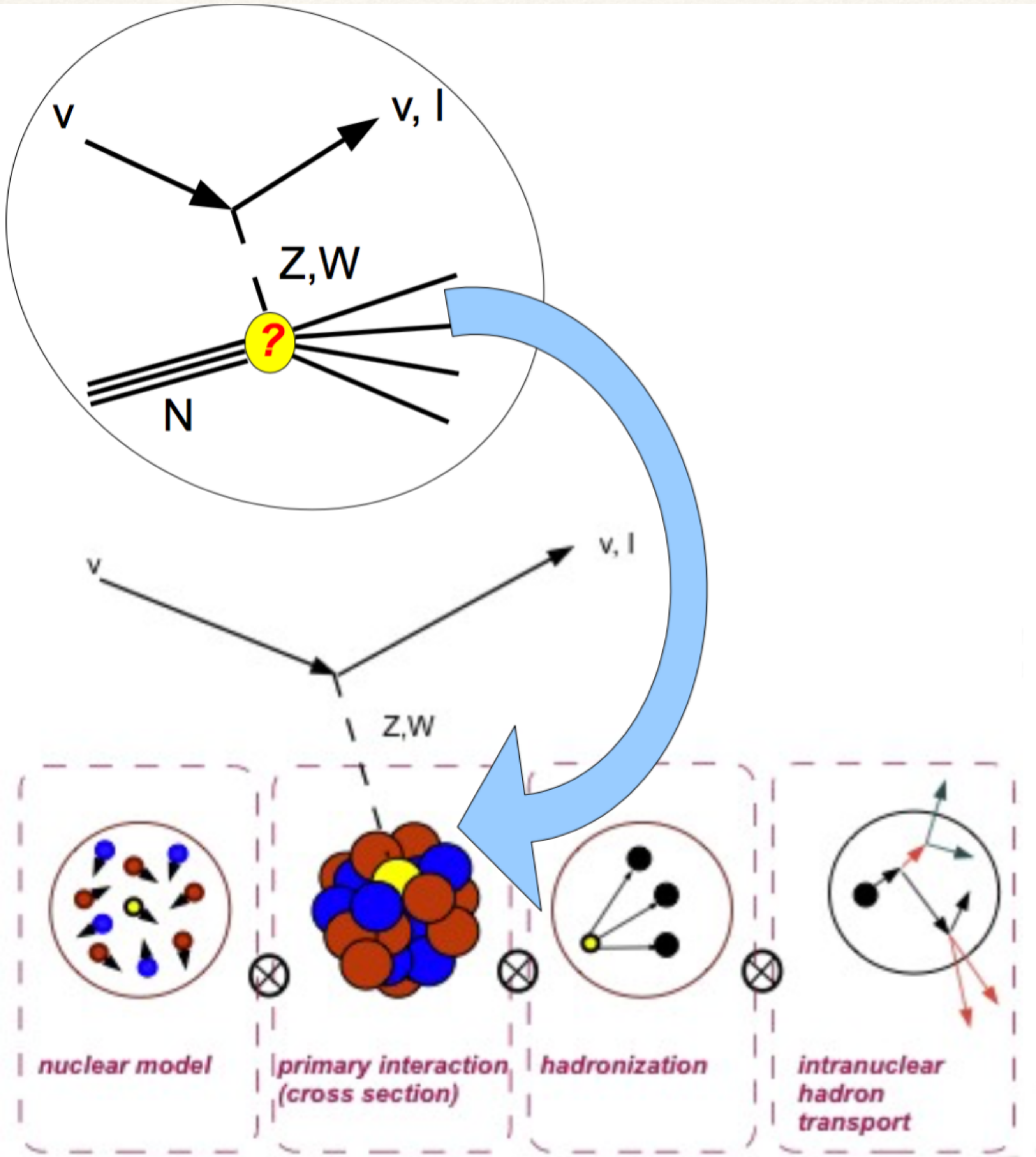


Stored flux metadata object specific to input flux driver

Extra tree branch
Event index links a GHEP leaf with flux metadata leaf

- Are ISI and FSI done at separate stages in your generator.
  - If not, why not?
  - How difficult, mechanically, would this factorization be?

Event generation factorisation in GENIE



Traditionally, GENIE has been using the generation factorisation shown on the left

FSI (intranuclear hadron transport) code is a separate step

This structure has been preserved for all current GENIE comprehensive model configurations, and it the same for all 4 GENIE FSI models:

- INTRANUKE / hA
- INTRANUKE / hN
- INCL
- GEANT

Event generation factorisation in GENIE

The primary hadronic system is clearly visible / separate from the final state (post-FSI) hadronic system and it is preserved in the event record

```
|--------------------------------------------------------------------------------------------------------------|
|GENIE GHEP Event Record [print level:   3]                                                                    |
|--------------------------------------------------------------------------------------------------------------|
| Idx |          Name | Ist |        PDG |   Mother  | Daughter  |      Px |      Py |      Pz |       E |       m |
|--------------------------------------------------------------------------------------------------------------|
|   0 |         nu_mu |   0 |         14 |  -1 |  -1 |   4 |   4 |   0.000 |   0.000 |   1.000 |   1.000 |   0.000 |
|   1 |          Ar40 |   0 | 1000180400 |  -1 |  -1 |   2 |   3 |   0.000 |   0.000 |   0.000 |  37.216 |  37.216 |
|   2 |       neutron |  11 |       2112 |   1 |  -1 |   5 |   5 |  -0.059 |   0.147 |  -0.183 |   0.929 | **0.940 | M = 0.897
|   3 |          Ar39 |   2 | 1000180390 |   1 |  -1 |  11 |  11 |   0.059 |  -0.147 |   0.183 |  36.287 |  36.286 |
|   4 |         nu_mu |   1 |         14 |   0 |  -1 |  -1 |  -1 |   0.210 |   0.195 |   0.476 |   0.555 |   0.000 | P = (-0.378,-0.351,-0.857)
|   5 |        Delta0 |   3 |       2114 |   2 |  -1 |   6 |   7 |  -0.269 |  -0.048 |   0.341 |   1.374 | **1.233 | M = 1.302
|   6 |       neutron |  14 |       2112 |   5 |  -1 |   8 |   8 |  -0.498 |  -0.050 |   0.291 |   1.103 |   0.940 | FSI = 1
|   7 |           pi0 |  14 |        111 |   5 |  -1 |   9 |  10 |   0.229 |   0.002 |   0.051 |   0.270 |   0.135 | FSI = 3
|   8 |       neutron |   1 |       2112 |   6 |  -1 |  -1 |  -1 |  -0.498 |  -0.050 |   0.291 |   1.103 |   0.940 |
|   9 |           pi0 |   1 |        111 |   7 |  -1 |  -1 |  -1 |   0.162 |   0.135 |   0.050 |   0.256 |   0.135 |
|  10 |       neutron |   1 |       2112 |   7 |  -1 |  -1 |  -1 |   0.154 |   0.066 |  -0.057 |   0.956 |   0.940 |
|  11 |      HadrBlob |  15 | 2000000002 |   3 |  -1 |  -1 |  -1 |  -0.029 |  -0.347 |   0.241 |  35.345 | **0.000 | M = 35.343
|--------------------------------------------------------------------------------------------------------------|
|       Fin-Init:                                                 |  -0.000 |  -0.000 |   0.000 |   0.000 |         |
|--------------------------------------------------------------------------------------------------------------|
|       Vertex:          nu_mu @ (x =     0.00000 m, y =     0.00000 m, z =     0.00000 m, t =    0.000000e+00 s) |
|--------------------------------------------------------------------------------------------------------------|
| Err flag [bits:15->0] : 0000000000000000     |  1st set:                                                 none |
| Err mask [bits:15->0] : 1111111111111111     |  Is unphysical:    NO |   Accepted:   YES                     |
|--------------------------------------------------------------------------------------------------------------|
| sig(Ev) =       2.68261e-38 cm^2  | d2sig(W,Q2;E)/dWdQ2 =     1.12848e-37 cm^2/GeV^3 | Weight =          1.00000 |
|--------------------------------------------------------------------------------------------------------------|
```

Caveats with hard-scattering / FSI separation

- **Not a true physics factorisation**
- Nuclear environment permeates all aspects of the event
  - Code structure for description of nuclear environment a next major upgrade in GENIE
  - This new structure should elevate the fact that this environment is common throughout the event
  - Probably will alter some model interfaces and may enforce pre/post-FSI nuclear model consistency
  - May limit scope for a combinatoric explosion of hard scattering / FSI models
- Pre-FSI simulation choices carefully matched to FSI capabilities
  - Example: FSI code doesn't handle propagation of Δ's, they are decayed before the FSI stage that handles its decay products
  - Example: Binding energy / bringing nucleons on the mass shell
- In medium effects to hadronization
  - Future upgrades will blur the boundary between the last 2 steps