G4Beamline

A “Swiss Army Knife” for Geant4

http://g4beamline.muonsinc.com

Tom Roberts
Muons, Inc.

Basic Approach

- Implement a general, flexible, and extensible program for Geant4 simulations, optimized for beamlines.
- In practice, it is much more general than just beamlines (e.g. there is a cosmic-ray “beam”)
- Requires no programming by users, but is sophisticated enough to simulate the Study II SFoFo Cooling Channel, and flexible enough to do the MICE beamline and cosmic-ray studies.
- Use a straightforward ASCII input file to completely determine the simulation.
- Provide user-friendly input generation, visualization of the system, and analysis of the simulation results.
- Include realistic and accurate particle tracking and interactions (incl. EM, weak, and hadronic).
- Include support for scripting and parallel jobs on a cluster

Using the Program

- The basic idea is to define each beamline element, and then place each one into the beamline at the appropriate place(s).
- All aspects of the simulation are specified in a single ASCII input file:
  - Geometry
  - Input Beam
  - Physics processes
  - Program control parameters
  - Generation of output NTuples
- The input file consists of a sequence of commands with named arguments
- Each command has its own list of arguments
- Command and argument names are spelled out, so the input file becomes a record of the simulation that is readable by others

Using the Program

- The beamline elements implemented are:
  - absorber – a material absorber with shaped containment and safety windows
  - box – a material in the shape of a box
  - corner – rotate the centerline coordinates, for bend or secondary target
  - cosmicraybeam – a “beam” of cosmic-ray muons
  - fieldmap – read a field map from a file, for E and/or B
  - genericbend – a generic bending magnet
  - genericquad – a generic quadrupole magnet
  - helicaldipole – a helical dipole magnet for 6-D muon cooling
  - idealsectorbend – a sector bending magnet
  - pillbox – a pillbox RF cavity, including optional windows
  - polycone – a material in the shape of multiple cones
  - solenoid – a single-coil magnet
  - sphere – a material in the shape of a sphere
  - trap – a material in the shape of a trapezoid
  - tubs – a material in the shape of a cylinder or pipe
  - virtualdetector – a ‘perfect’ detector for monitoring the beam

Using the Program

- Simulation control commands:
  - beam – specify the incoming beam (from a file or randomly generated)
  - reference – specify a reference particle
  - place – position a previously-defined object into the simulation
  - material – specify the properties of a new material
  - geometry – perform geometrical tests for invalid intersections of objects
  - param – define parameters for program or input file
  - particlecolor – specify the display colors for particle types
  - particlefilter – cut particles by type or momentum, force decays, etc.
  - physics – defines the physics processes and controls them
  - trackcuts – impose specific cuts on tracks
- Beamline layout commands:
  - start – defines the starting point and orientation of the beamline
  - corner – inserts a corner into the Centerline coordinates
  - cornerarc – inserts a corner into the Centerline coordinates, with path length of an arc

Using the Program

- Complex manual procedures have been automated:
  - Field maps of solenoids can be automatically determined by specifying the required accuracy
  - RF Cavities can be tuned, for timing and gradient; both can be fixed or automatically tuned
- Geometry layout has been vastly simplified
  - Beam elements are simply lined up along the Z axis
  - Centerline coordinates behave naturally for bending magnets or secondary targets
  - Elements may overlap (e.g. nested pipes), but not intersect
  - Many elements can be the parent of other elements
  - Specific offsets in X, Y, and/or Z can be specified when needed
  - Rotations are specified naturally
    - Y30,Z90 is a 30 degree rotation around Y followed by a 90 degree rotation around Z
    - Axes are for the <u>parent</u> volume and thus do not change
  - Automatic geometry testing detects invalid intersections
- All of the Geant4 8.1 physics use cases are available by name.
- Beam tracks can be generated internally, or read from a file
- Most Geant4 visualization drivers are supported by name.
  - Open Inventor is included, and is by far the most user friendly

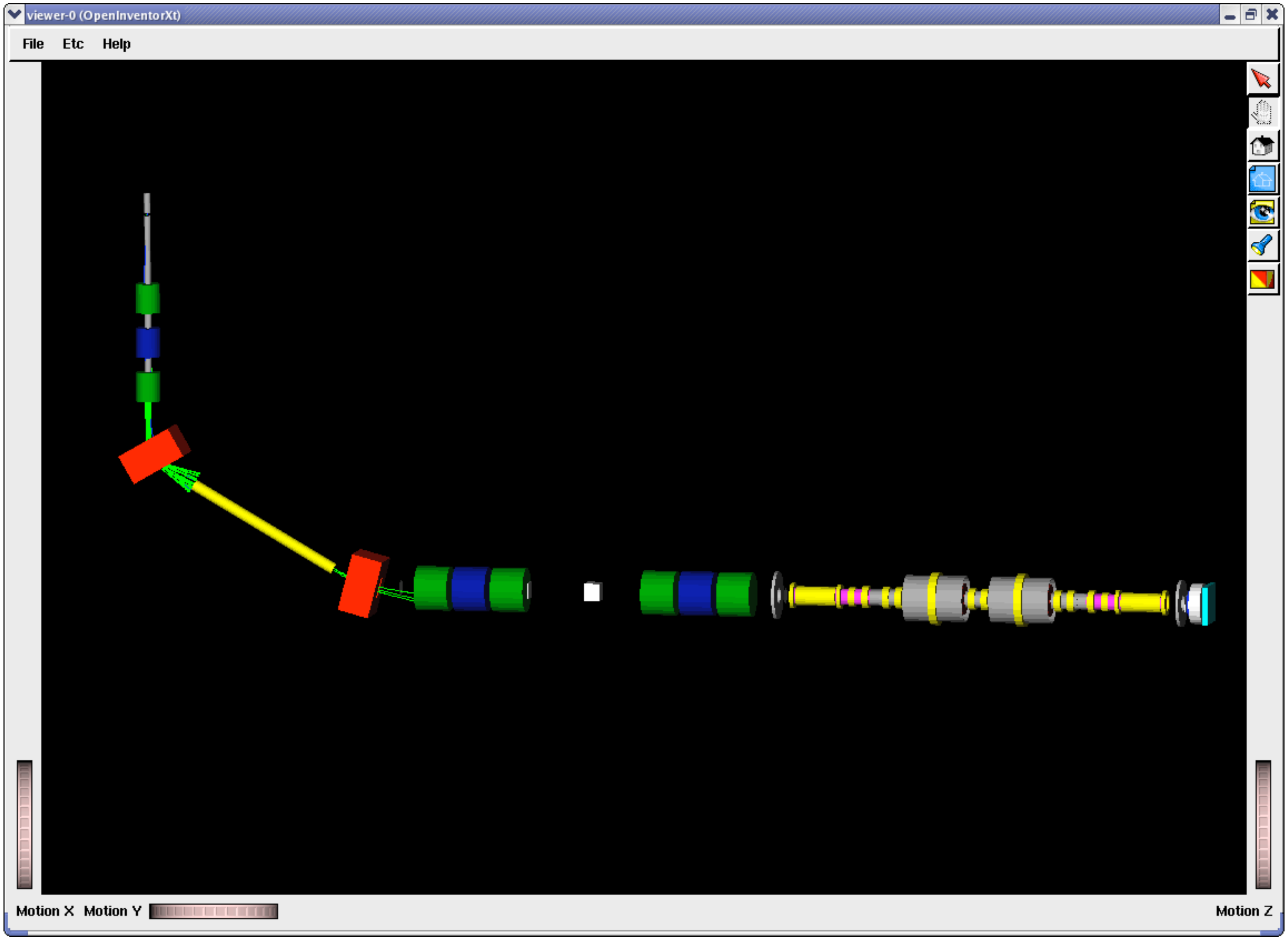
Using the Program

- The result is a program that reduces the complexity of the user input to that of the system being simulated (a major drawback of any simulation program, including those using Geant4, is that C++ simulation code is considerably more complex than the problem).

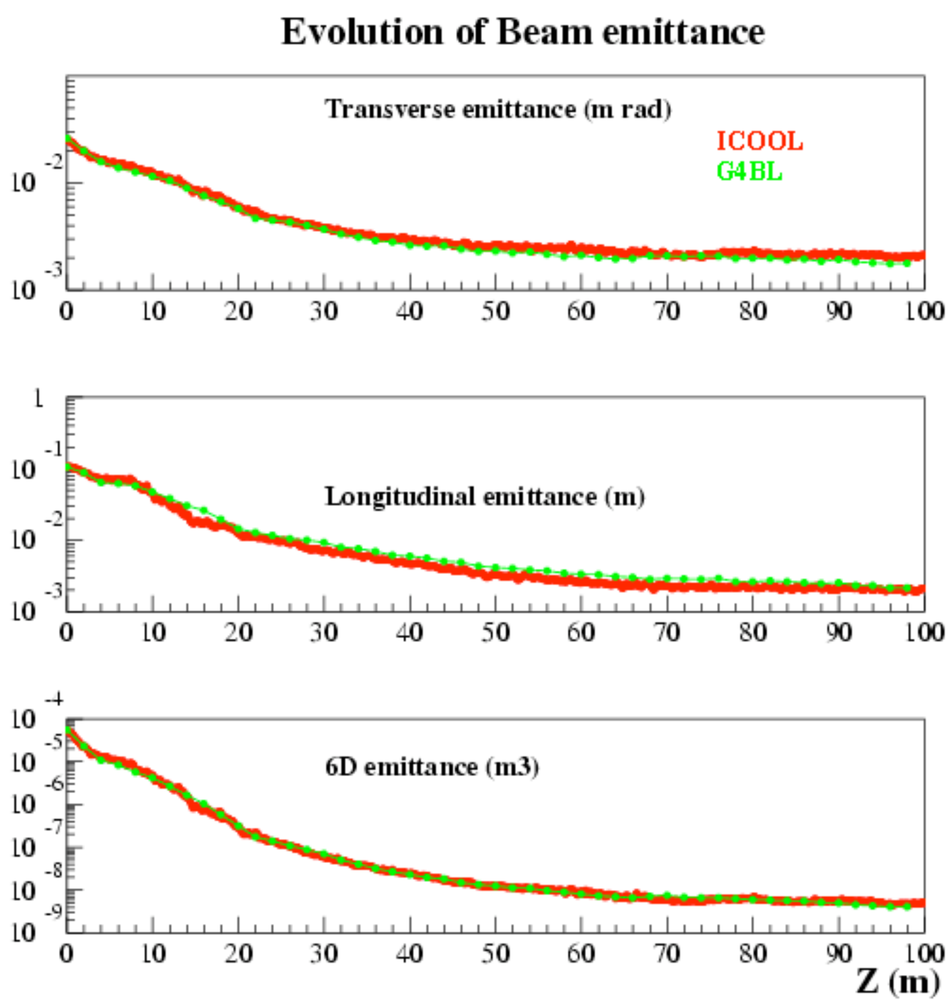
- While C++ programming is not required to use the program, knowledge of the problem domain is absolutely required, as is enough experience to distinguish sensible results from nonsense.

- Visualization is <u>highly</u> recommended, to verify that the geometry is correct and makes sense.

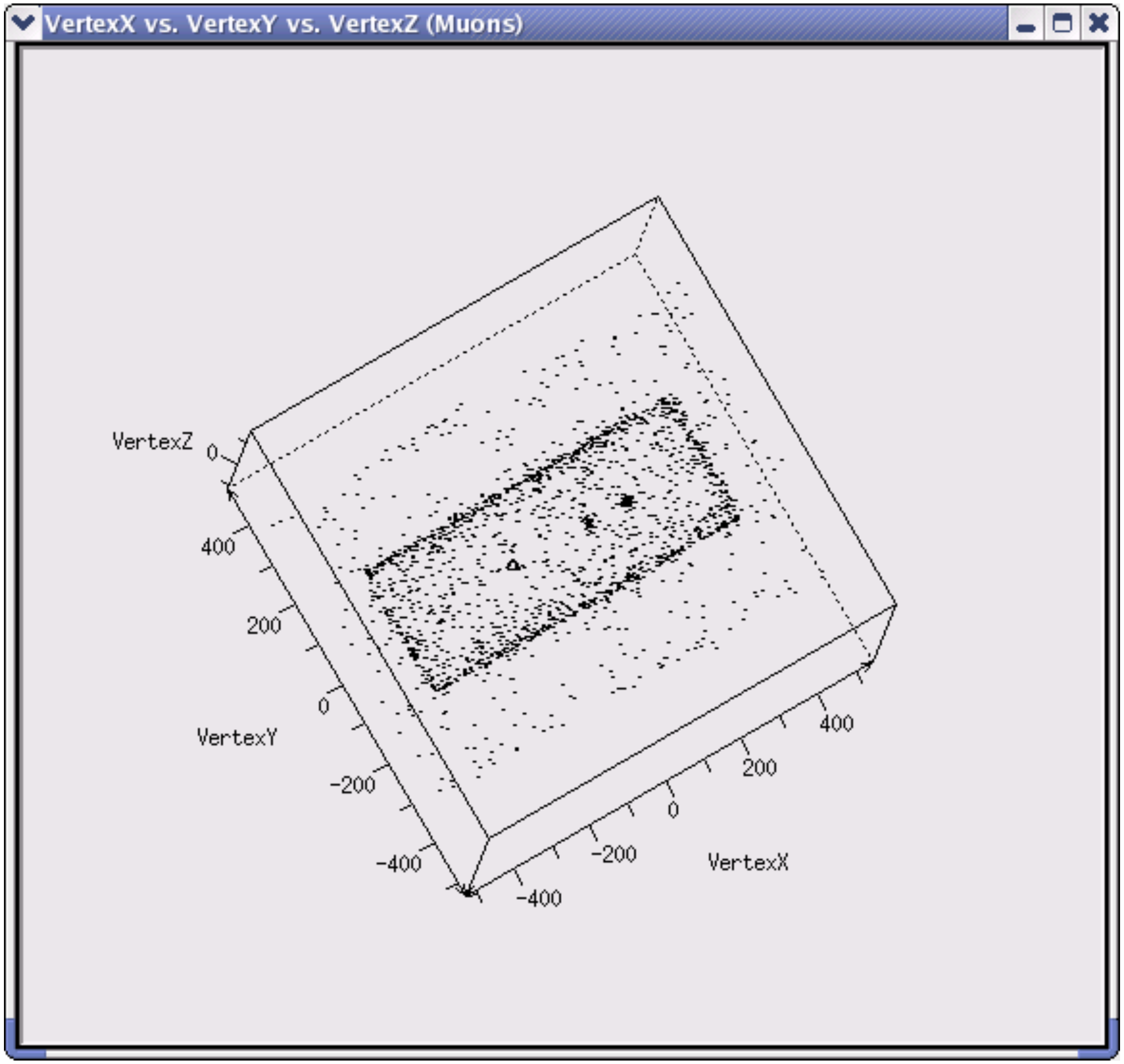
Example Uses – MICE



Example Uses – 6D Muon Cooling



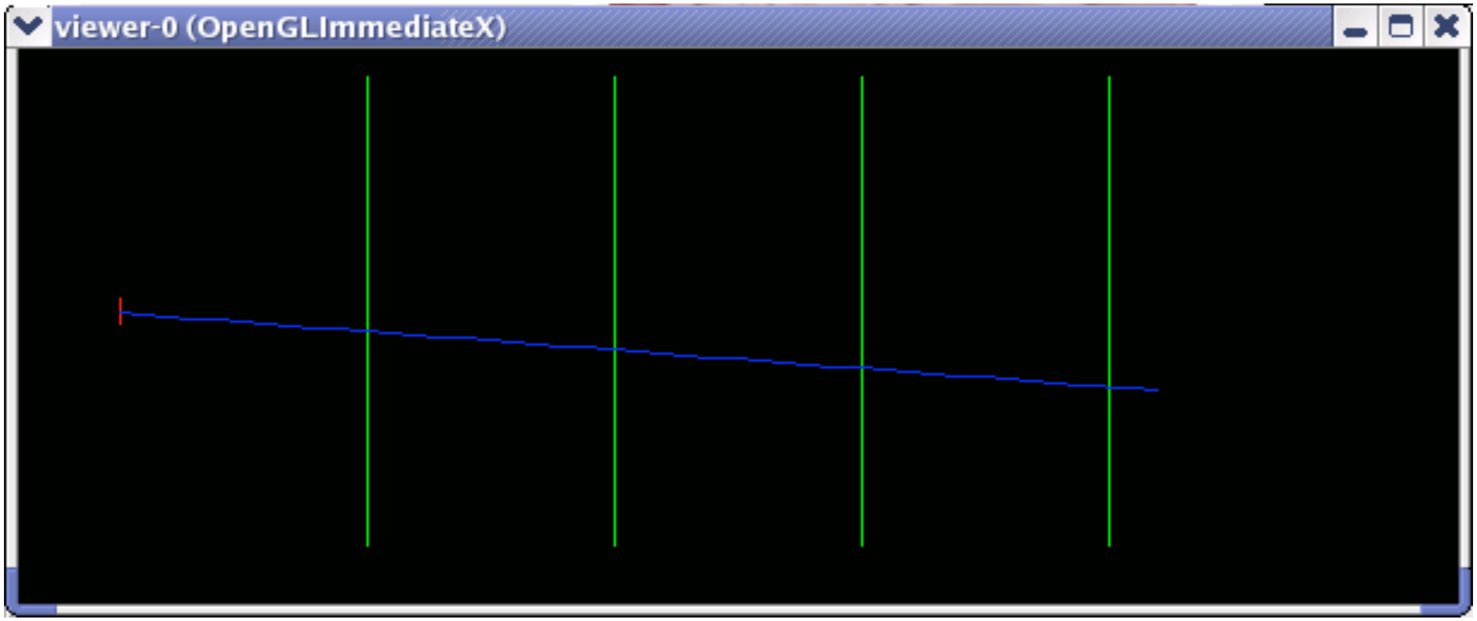
Example Uses – Cosmic Ray Tomography



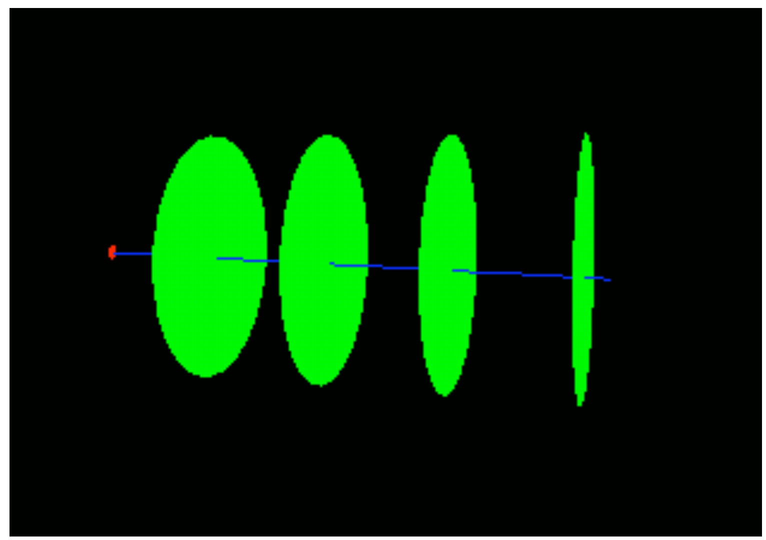
example1.in

```
# example1.in – put beam into 4 detectors
physics  LHEP_BIC
beam  gaussian  particle=mu+  nEvents=1000  beamZ=0.0 \
     sigmaX=10.0  sigmaY=10.0  sigmaXp=0.100  sigmaYp=0.100 \
     meanMomentum=200.0  sigmaP=4.0  meanT=0.0  sigmaT=0.0
# BeamVis just shows where the beam comes from
Box  BeamVis  width=100.0  height=100.0  length=0.1 color=1,0,0
# define the detector (used 4 times)
detector  Det  radius=1000.0  color=0,1,0
# place BeamVis and four detectors, putting their number into their names
place  BeamVis  z=0
place  Det  z=1000.0  rename=Det#
place  Det  z=2000.0  rename=Det#
place  Det  z=3000.0  rename=Det#
place  Det  z=4000.0  rename=Det#
```

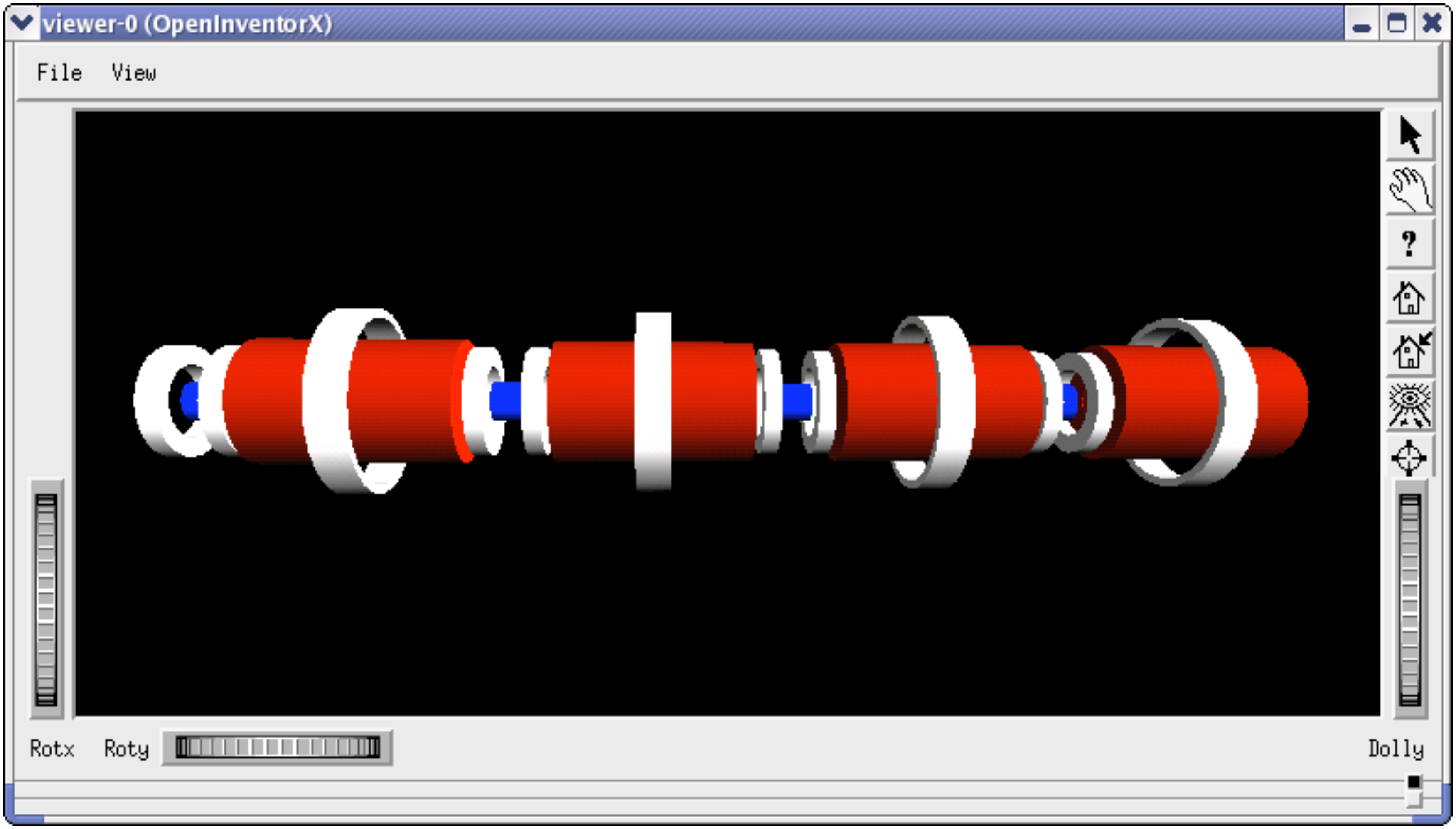
example1.in



Visualizations of example1.in, using OpenGL (above), and OpenInventor (right). Above is a plan view, while at right is a 3-D view.



example2.in (4 cells from Study 2)



Demonstration of interactive capabilities

- Visualization of the MICE beamline via Open Inventor
- Generation of histograms and manipulating them via HistoRoot

Summary

- G4Beamline is a simulation program capable of accurate and realistic simulation via single-particle tracking.
- It has an intuitive, user-friendly interface that reflects the complexity of the problem, and is directly readable by physicists familiar with the problem domain.
- Simulations of complex accelerator structures can be performed without C++ programming.

http://g4beamline.muonsinc.com