“Geant4 for Medical Education”
Part 2
Toolkit and Courseware

Geant4 Workshop 2006
25 January 2006
Kitasato University

Hajime Yoshida
Naruto University of Education

Geant4 for Education Workshop

September 12 – 16 , Naruto University of Education, sponsored by KEK program
http://erpe1.naruto-u.ac.jp/~geant4/

Objective : Not to teach Geant4 but to use it to teach

Participants = Developers and course material creators

Name	affiliation	category	fields/backgrounds
Michel Maire	LAPP, Annecy	Geant4 developer	HEP
Dennis Wright	SLAC, Stanford	Geant4 developer	HEP
Koichi Maruyama	Kitasato Univ.	Geant4 user	medical sciences
Tomoyuki Hasegawa	Kitasato Univ.	Geant4 user	radiological technology
Katsuya Amako	KEK	Geant4 developer	HEP
Takashi Sasaki	KEK	Geant4 developer	HEP
Koichi Murakami	KEK	Geant4 developer	HEP
Go Iwai	JST/KEK	Geant4 user	HEP
Satoshi Kameoka	JST/KEK	Geant4 user	Nuclear Physics
Hajime Yoshida	Naruto UE	Geant4 developer	HEP
Yoshihiro Kawanishi	Naruto UE	Geant4 UI	Technology Education

Requirements on Geometry

- Realistic and Standard
  - Concrete and realistic “standard” geometries must be provided whose geometrical data must be available publicly.
- Generic and Customizable
  - Some generic geometries which can be customized by teachers will be useful to create their own course ware
- Interactivity
  - We need much more interactivities for creators of courseware to customize for their own applications

Realistic and Standard Geometries - 1 Medical

- Standard ionization chambers
  - Track visualization in and around
  - Build up cap
  - Total number of created ions
- Curie well chamber
- Gamma camera
  - Number of photons
  - Energy spectrum
- PET
- GM counter
  - Track visualization

Generic Geometries - 2
- HEP and NP

- Tracking
  - Vertex chamber
  - Drift chamber
  - TPC
  - MWPC
- Calorimetry
  - Sampling energy and position
  - Crystals
- PID
  - Cerenkov
  - ToF
  - dE/dX
- Absorbers

Physics lists for medical

- A common physics list must be provided -> done!.
- Medical max < 1 GeV
- Start with N03
  - Switching on/off any processes
  - Hadronic processes
    - P elastic, inelastic
    - N elastic, inelastic
    - π
    - Ion
    - Radioactive decays, generic decays
  - Choice of models
    - LEP, Bertine, Binary cascade
  - Process can be turned on one by one. Range cut and step size must be easily modifiable
  - Only the hadronic processes can be visualized – Michel's magic

Physics list for HEP/NP

- We must provide a pre-built list each of whose processes can be turned on/off one by one
- Fixed models are to be used
- Can control the number of secondaries

Geant4Py Toolkits for Educational Applications

- We should take care of two user categories;
  - Contents Creators (*teachers*)
  - End Users (*students*)
- Geant4Py Toolkits For Contents Creators
  - Developed by K. Murakami, now in beta version available from CVS
  - Python's powerful scripting capabilities are exploitable
  - Python interface can work as component bus.
  - Modularizing, combining, and using components
    - Material / Geometry (predefined geometry / easy geometry setup)
    - Physics list (EM, Hadrons, Ion)
    - Detector response (Calorimeter / Tracker)
    - Analysis packages (ROOT, HBOOK, AIDA, ...)
    - Visualization
    - GUI (Qt, Tkinter, ...) / Web applications (mod-python, CherryPy)
- Courseware For End Users
  - Scripting with Python is NOT essential!
    - Of course, they can play with scripting.
    - They are not necessarily required to learn Python language.
  - GUI / Web applications should be presented.
    - They can be built on the Python interface.

Geant4Py
List of Exposed Classes

- **global**
  - G4String
  - G4ThreeVector
  - G4RotationMatrix
  - ...
- **interface**
  - G4UImanager
  - G4UIterminal
- **run**
  - G4RunManager
  - G4VUserDetectorConstruction
  - G4VUserPhysicsList
  - G4UserRunAction
  - G4VUserPrimaryGeneratorAction
  - G4Run
  - ...
- **event**
  - G4Event
  - G4ParticleGun
  - G4UserEventAction
  - ...
- **tracking**
  - G4UserSteppingAction
  - ...
- **track**
  - G4Step
  - G4Track
  - G4StepPoint
  - G4StepStatus
  - G4TrackStatus
  - ...
- **particles**
  - G4ParticleDefinition
  - G4DynamicParticle
  - G4PrimaryParticle
  - G4PrimaryVertex
  - ...
- **geometry**
  - G4VTouchable
  - G4TouchableHistotry
  - G4VPhysicalVolume
  - ...
- **material**
  - G4Material
  - G4NistManager
  - ...
- **visualization**
  - G4VisManager
  - G4VGraphicSystem
  - G4OpenGLStoredX
  - ...

Predefined packages

- We will also provide site-module package as pre-defined components.
  - Material
    - sets of pre-defined materials
      - NIST materials via G4NistManager
  - Geometry
    - “exN03” geometry as an example of pre-defined geometries
    - *“EZgeometry”*
      - provides functionalities for easy geometry setup (applicable to target experiments)
  - Physics List
    - pre-defined physics lists
    - easy access to cross sections, stopping powers, ... via *G4EmCalculator*
  - Primary Generator Action
    - particle gun
  - Sensitive Detector
    - calorimeter type
    - tracker type
- They can be used just by importing modules.
- They can be combined and connected to higher application layers (Analysis / GUI components).

Ezgeom

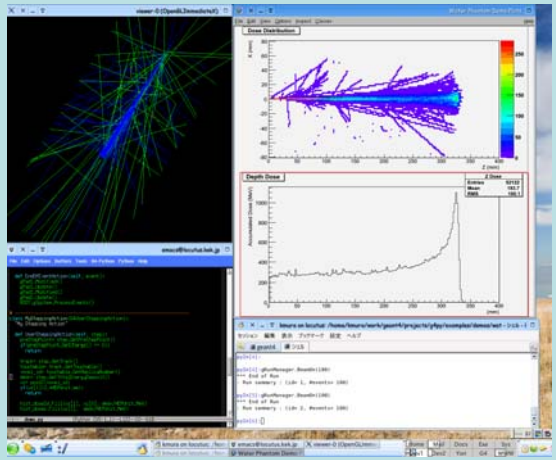
- class G4EzVolume
  - Construct, ResetWorld, ResizeWorld, SetWorldMaterial, SetWorldVisibility
  - CreateBox/Tube/Cone/Sphere/OrbVolume, SetSold, GetSold, SetMaterial, GetMaterial, PlaceIt, ReplicateIt, VoxelizeIt, SetSensitiveDetector, SetColor, SetVisibility
- usage
  - import Geant4
  - from Ezsim import Ezgeom
  - from Ezsim.EZgeom import EzG4Volume
  - alminium = G4Material.GetMaterial(“G4_Al”,1)
  - myLV = G4EzVolume(“myLogicalVolume”)
  - myLV.CreateBoxVolume(alminium, 1.*cm, 1.*cm, 1.*cm)
  - water=G4Material.GetMaterial(“G4_WATER”,1)
  - myLV.SetMaterial(water)
  - myPV=myLV.PlaceIt(G4ThreeVector(0.,0.,0.))
  - G4EzVolume.GetSold(myLV).SetXHalfLength(1.*m)
  - Ezgeom.ResizeWorld(2.*m, 1.*m, 1.*m)
  - myPV.SetTranslation(G4ThreeVector(50.*cm, 0.,0.))
  - myLV.SetColor(1.,0.,0.)

Ezgeom examples

- g4py/tests/gtest02/test.py
  - Just as is defined in exampleN03 C++ class
  - It is exposed with its methods
    - exN03geom.Get/Set methods for example
- g4py/tests/gtest03/test_voxel.py
  - VoxelizeIt(10,10,10)
- g4py/tests/iwai_demo/Em3.py
  - ReplicateIt()
- g4py/examples/demos/water_phantom/demo.py
  - Voxelized geometry is defined in C++ and exposed
  - Scoring action is scripted, using exposed objects

proton into the water phantom

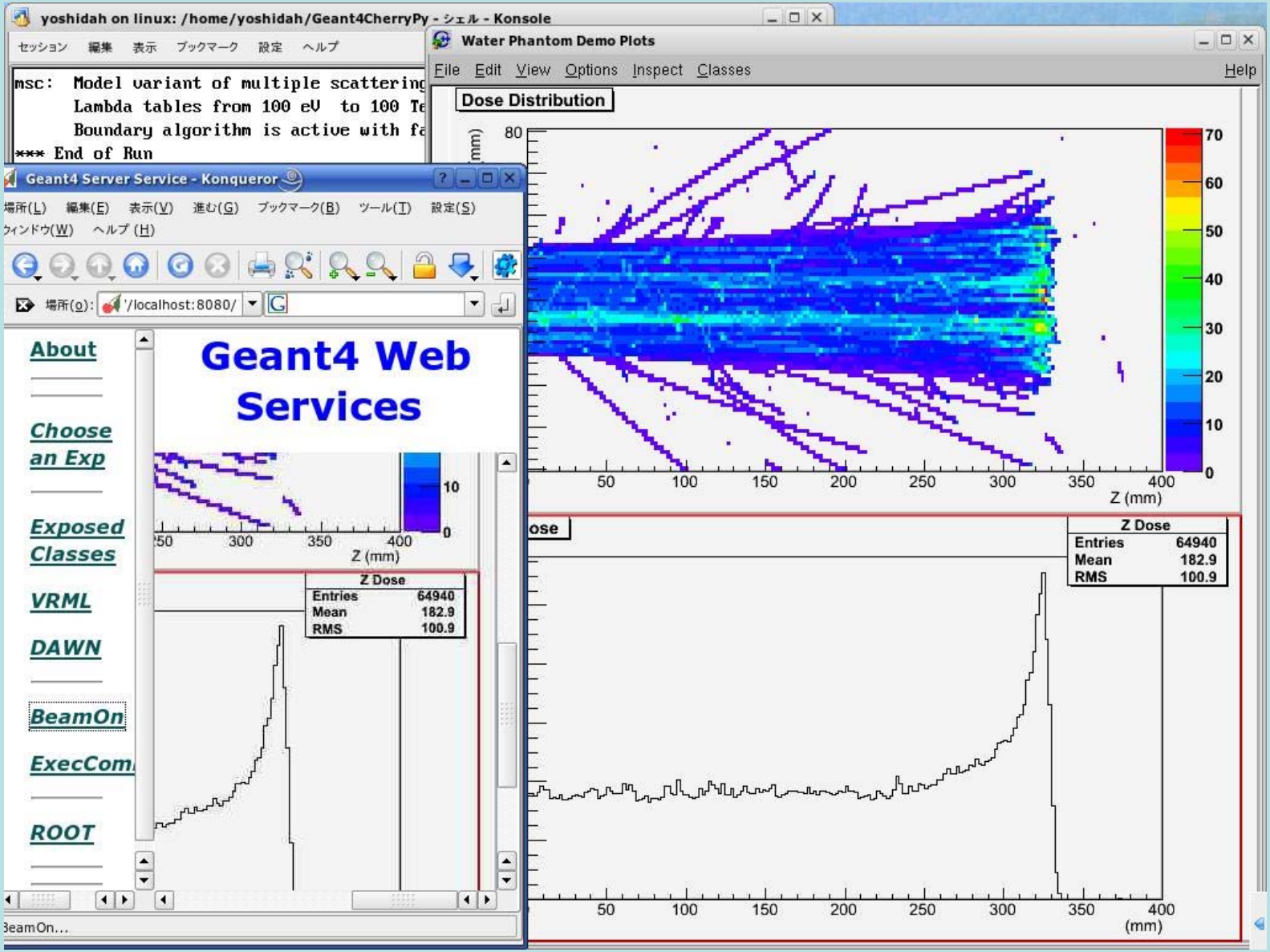
- An example of “water phantom dosimetry”
  - *This demo program shows that a Geant4 application well coworks with ROOT on the Python front end.*
- You can look features of;
  - dose calculation in a water phantom
  - Python implementation of sensitive detector
  - Python overloading of user actions
  - on-line histogramming with ROOT
  - visualization



Ezsim and ROOT

```
class ScoreSD(G4VSensitiveDetector):
  "SD for score voxels"
  def __init__(self):
    G4VSensitiveDetector.__init__(self, "ScoreVoxel")
  def ProcessHits(self, step, rohist):
    preStepPoint= step.GetPreStepPoint()
    if(preStepPoint.GetCharge() == 0):
      return
    track= step.GetTrack()
    touchable= track.GetTouchable()
    voxel_id= touchable.GetReplicaNumber()
    dedx= step.GetTotalEnergyDeposit()
    xz= posXZ(voxel_id)
    hist_dose2d.Fill(xz[1], xz[0], dedx/MeV)
    if( abs(xz[0]) <= 100 ):
      hist_dosez.Fill(xz[1],  dedx/MeV)
...
...
scoreSD= ScoreSD()
myDC.SetSDtoScoreVoxel(scoreSD)
```

Geant4 Web Service



ToDo list

- Geant4Py
  - Beta testing
  - To be distributed in the Geant4 package
- Toolkit
  - GUI framework
  - Generic geometries
- Courseware
  - For some typical use cases
- Packaging
  - Geant4 + Geant4Py + coursewares