GEant4 **M**onte**C**arlo

geant4

esempio gemc

panoramica gemc

Maurizio Ungaro, Jefferson Lab

Geant4 Toolkit

- Nuclear Physics experiments and detector design
- Radiation shielding (in space too)
- Calorimetry
- Cosmic rays
- Neutrino physics
- Dosimetry
- Radiotherapy
- Biological damage studies
- Assessment of radiation damage to the electronics of satellites
- Study of the radiation environment of planets.

Geant4 Toolkit

Basic Example B5

Hardcoded Numbers
Hardcoded Instances names
Other hardcoded variables

```
// hodoscopes in first arm
auto hodoscope1Solid
   = new G4Box(“hodoscope1Box",5.*cm,20.*cm,0.5*cm);
```

→ Solid

```
fHodoscope1Logical
   = new
G4LogicalVolume(hodoscope1Solid,scintillator,"hodoscope1Logical");
```

→ Logical

```
for (auto i=0;i<kNofHodoscopes1;i++) {
    G4double x1 = (i-kNofHodoscopes1/2)*10.*cm;
    new
G4PVPlacement(0,G4ThreeVector(x1,0.,-1.5*m),fHodoscope1Logical,
                         "hodoscope1Physical",firstArmLogical,
                         false,i,checkOverlaps);
}
```

→ Physical

Problemi

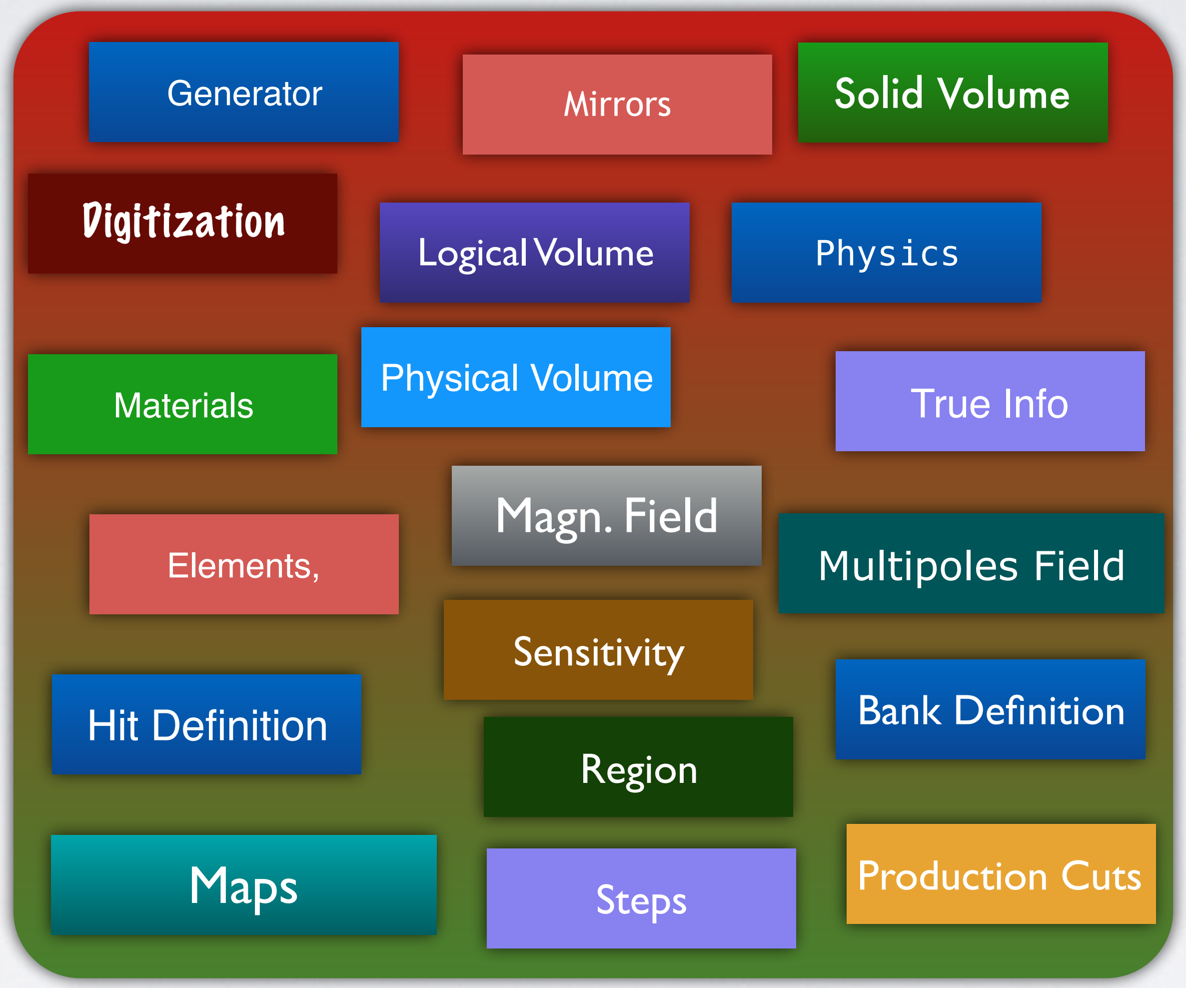
E' necessaria la conoscenza (C++, Geant4) di:

- Come e dove si costruisce un volume solido, logico, fisico
- Come e dove si definisce un volume "attivo"
- Come e dove si trattano i vari hits nei vari volumi attivi.
- Come e dove si organizza e si scrive l'output.
- Come e dove si definiscono i campi magnetici.

Come tutte queste variabili e numeri interagiscono tra di loro!

Hard-coded numbers, hardcoded variables names: e' un incubo!

Simulazione Realistica

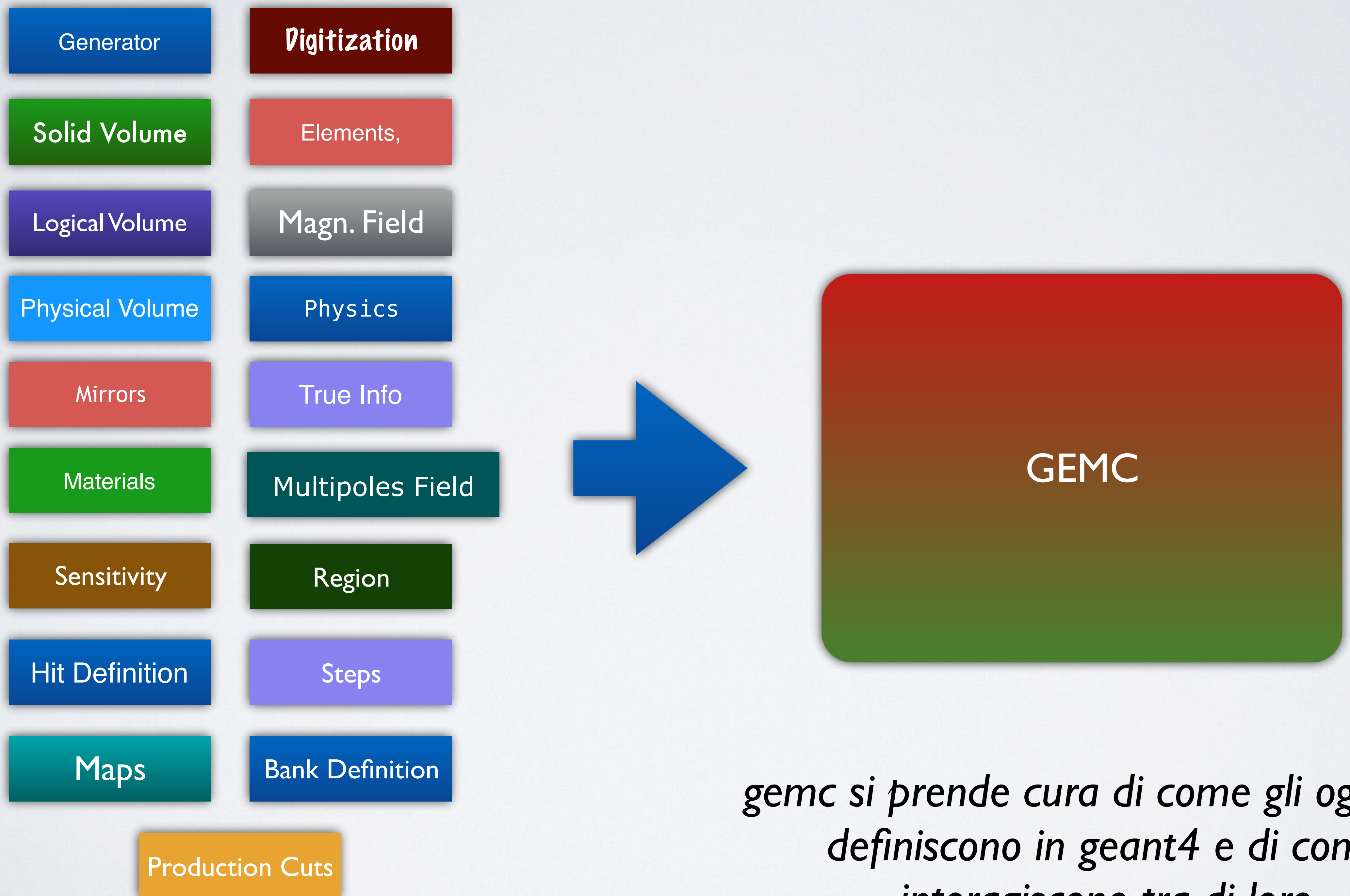


Ingredienti per una simulazione realistica

Dobbiamo anche scrivere la comunicazione tra gli oggetti in ognuno di questi ingredienti e gli oggetti negli altri ingredienti

Simulazione Realistica

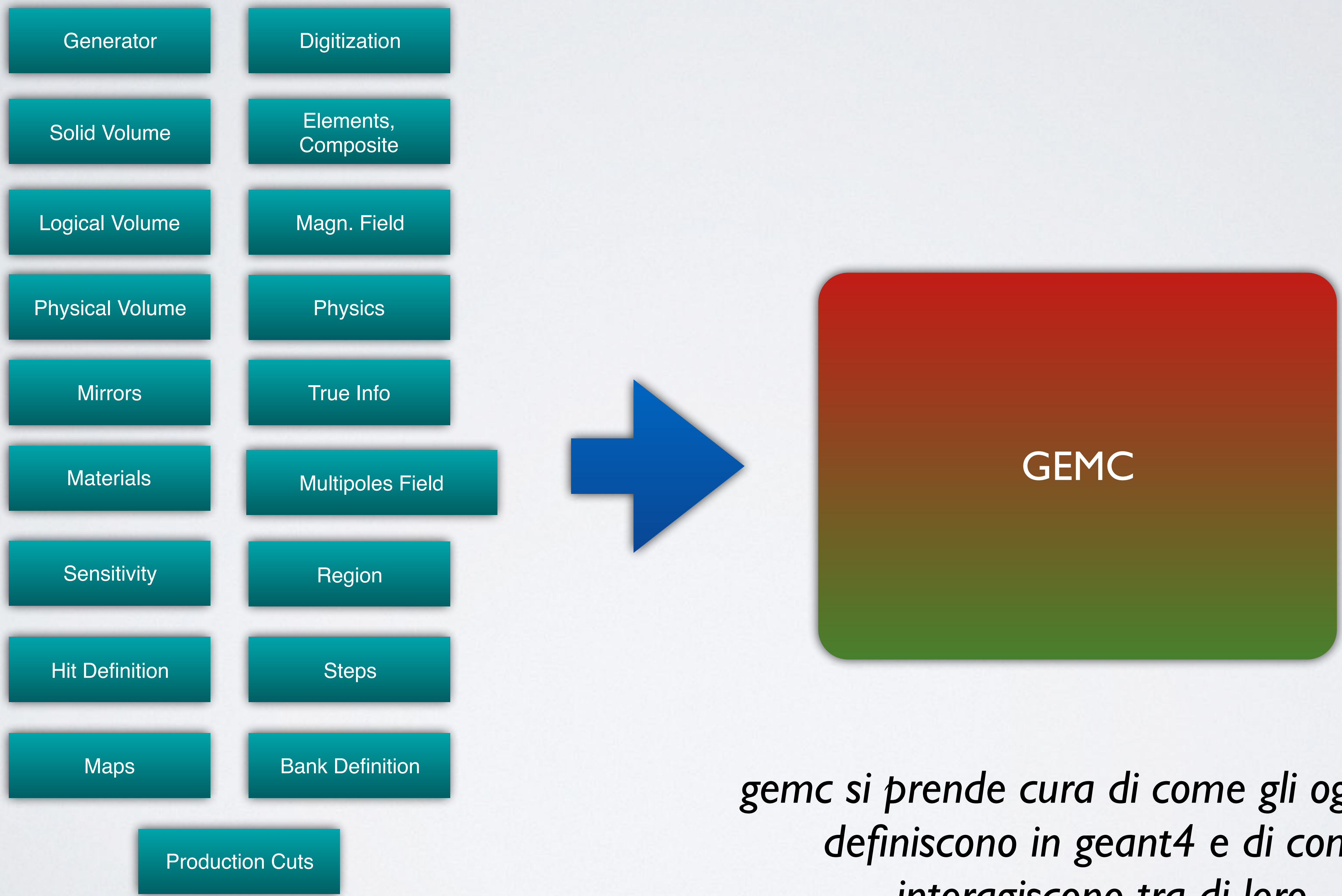
abstraction of "detector" to table of parameters



gemc si prende cura di come gli oggetti si definiscono in geant4 e di come interagiscono tra di loro

Simulazione Realistica: GEMC

standardized api for all components



gemc si prende cura di come gli oggetti si definiscono in geant4 e di come interagiscono tra di loro

parametri in databases

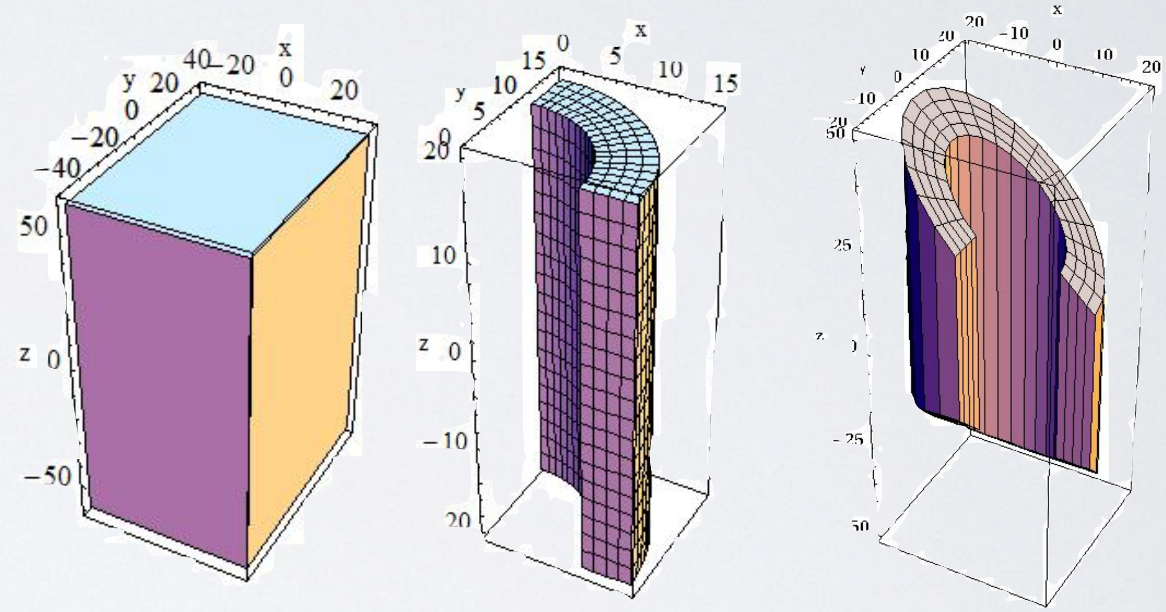
Esempio: definizione di volume in geant4

```
def makeGeometry(configuration):

detector = MyDetector(name="paddle_01", mother="root")
detector.description  = "Si detector"

detector.type        = "Tube"
detector.dimensions = "0.*cm 1.*cm 5.*mm 0*deg 360*deg"

detector.material = "G4_Si"
```

Da sapere:

<u>che volumi sono a disposizione</u>

<u>che materiali sono a disposizione</u>



Z	Name	ChFormula	density(g/cm^3)	I(eV)
1	G4_H		8.3748e-05	19.2
2	G4_He		0.000166322	41.8
3	G4_Li		0.534	40
4	G4_Be		1.848	63.7
5	G4_B		2.37	76
6	G4_C		2	81

Esempio: attributi grafici

```
def makeGeometry(configuration):

detector = MyDetector(name="paddle_01", mother="root")
detector.description  = "Si detector"

detector.type        = "Tube"
detector.dimensions = "0.*cm 1.*cm 5.*mm 0*deg 360*deg"

detector.material = "G4_Si"

detector.visible     = 1            # 1 visible, 0 to leave hidden
detector.style       = 1            # 1 displays as a solid, 0 as wireframe
detector.color       = “f4a988"
```

Vantaggi:

- E’ uno script. Non c’e’ bisogno di ricompilare.
- I numeri possono venire da un DB.
- Le “cose” da sapere di geant4 sono 2: nomi dei solidi, e nomi dei materiali.

Esempio: sensitività, digitizzazione

```
def makeGeometry(configuration):

detector = MyDetector(name="paddle_01", mother="root")
detector.description  = "Si detector"

detector.type       = "Tube"
detector.dimensions = "0.*cm 1.*cm 5.*mm 0*deg 360*deg"

detector.material = "G4_Si"

detector.visible     = 1                  # 1 visible, 0 to leave hidden
detector.style       = 1                  # 1 displays as a solid, 0 as wireframe
detector.color       = “f4a988"

detector.sensitivity = "flux"             # Use the “flux” sensitivity. Defines the output
detector.hit_type    = "flux"             # Use the “flux” digitization: defines response.
detector.identifiers = "paddle manual 1"  # Identifies the detector being hit
```

Vantaggi:

- E’ uno script. Non c’e’ bisogno di ricompilare.
- I numeri possono venire da un DB.
- Le “cose” da sapere di geant4 sono 2: nomi dei solidi, e nomi dei materiali.
- Tutte le definizioni necessarie sono compatte, in un solo posto.

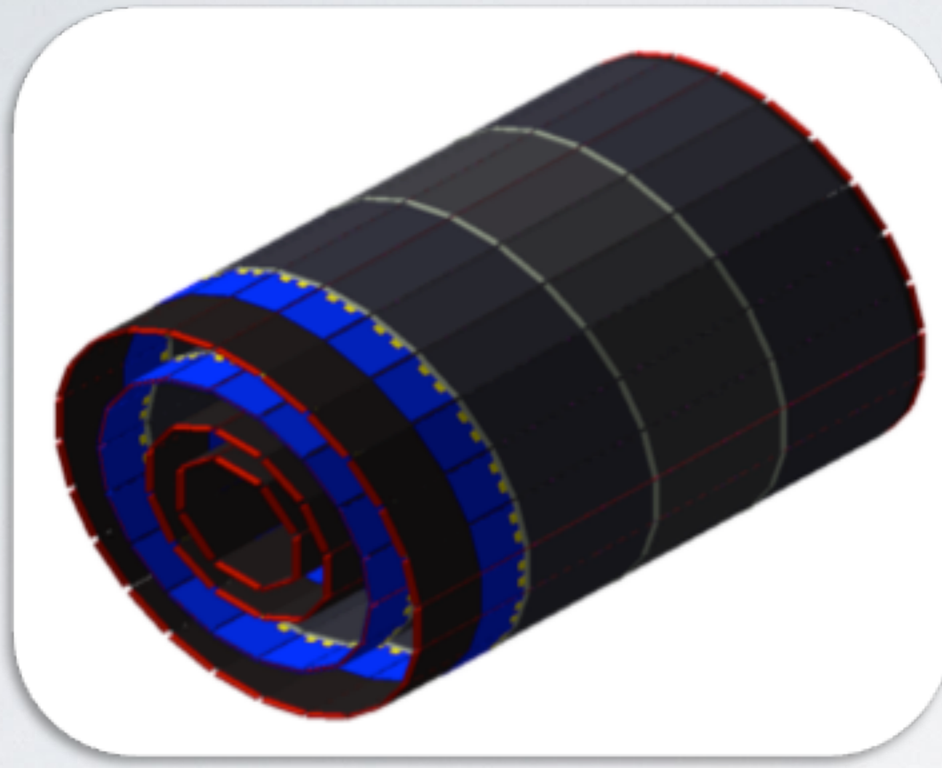
Esempio: campo magnetico

```
def makeGeometry(configuration):

detector = MyDetector(name="paddle_01", mother="root")
detector.description  = "Si detector"

detector.type        = "Tube"
detector.dimensions = "0.*cm 1.*cm 5.*mm 0*deg 360*deg"

detector.material = "G4_Si"
detector.mfield   = "Torus"

detector.visible     = 1                  # 1 visible, 0 to leave hidden
detector.style       = 1                  # 1 displays as a solid, 0 as wireframe
detector.color       = “f4a988"

detector.sensitivity = "flux"             # Use the “flux” sensitivity. Defines the output
detector.hit_type    = "flux"             # Use the “flux” digitization: defines response.
detector.identifiers = "paddle manual 1" # Identifies the detector being hit
```

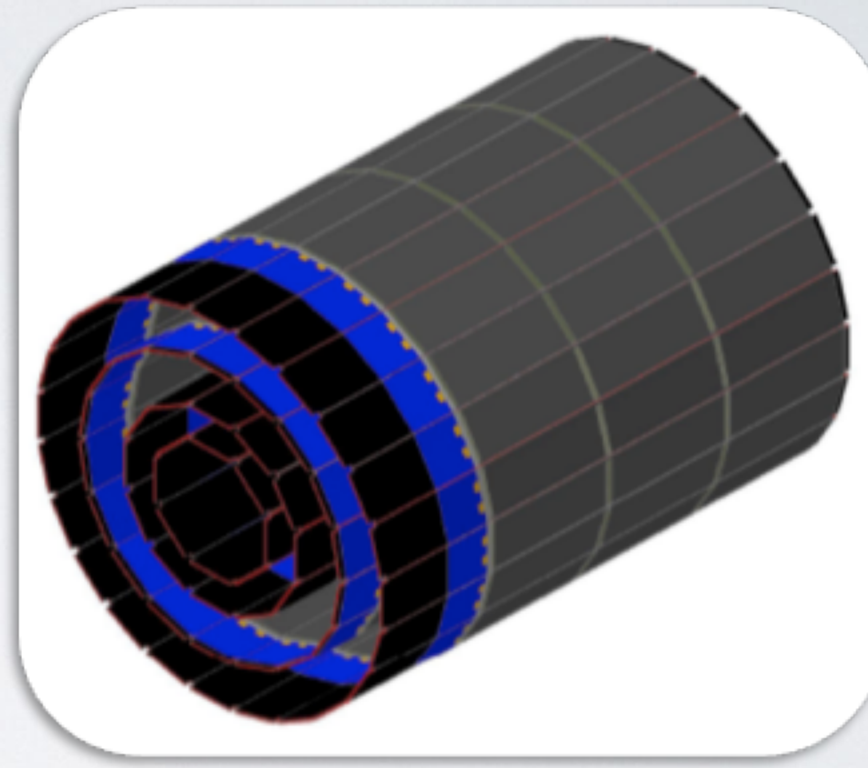
Vantaggi:

- E’ uno script. Non c’e’ bisogno di ricompilare.
- I numeri possono venire da un DB.
- Le “cose” da sapere di geant4 sono 2.
- Tutte le definizioni necessarie sono compatte, in un solo posto.
- GEMC si cura di tutto il resto: digitizzazione, campi magnetici, output, etc.

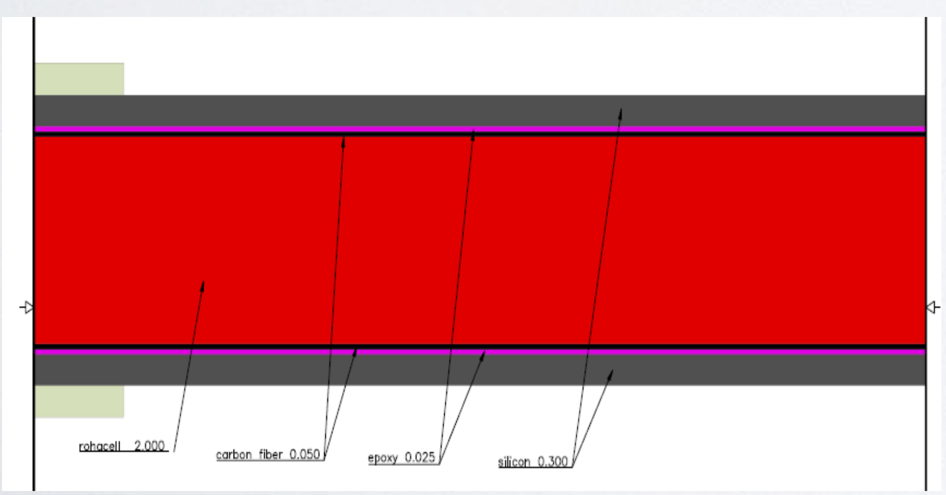
Esempio di un detector usato a Jefferson Lab

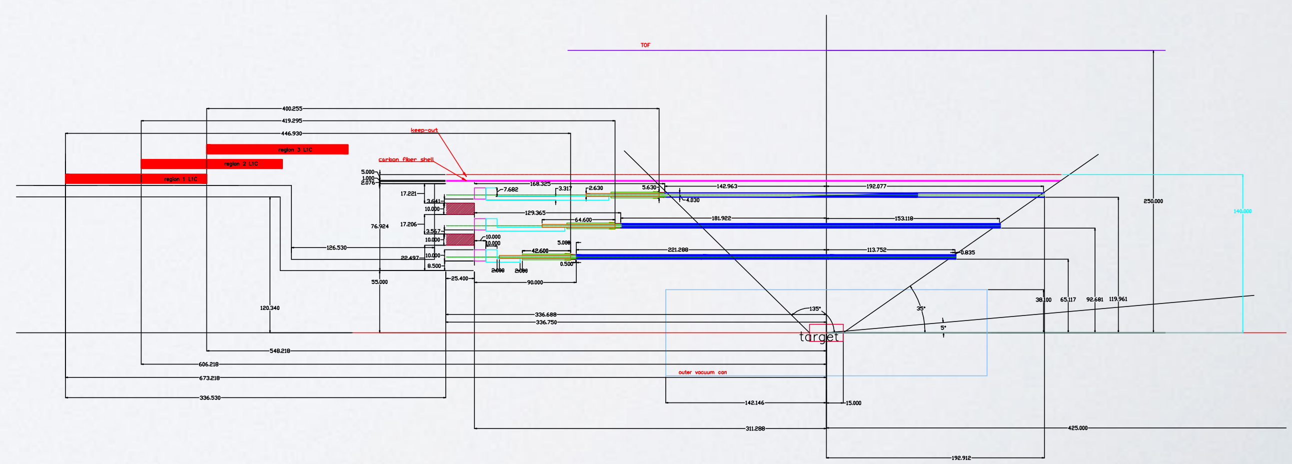


GEMC

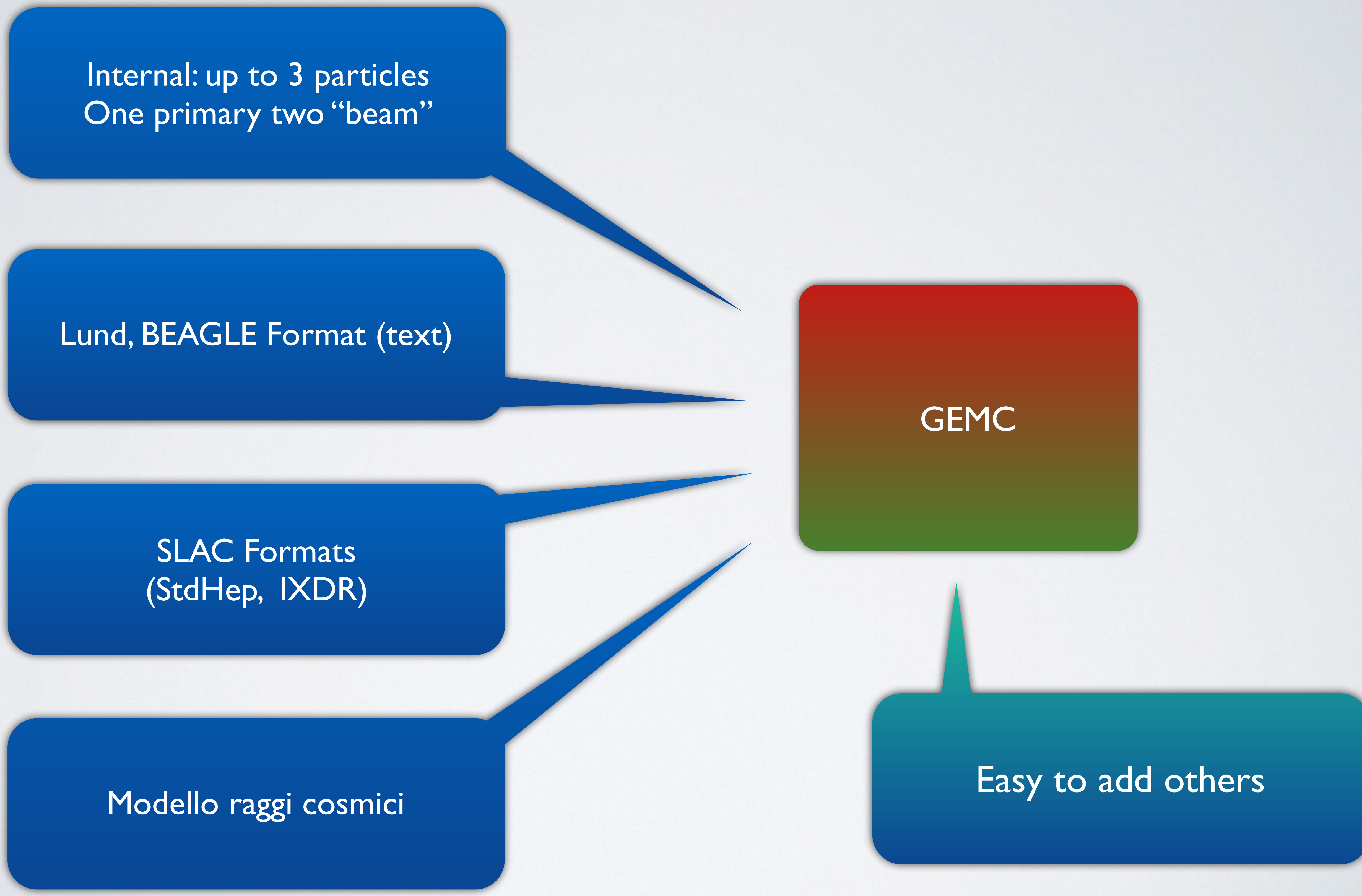


Engineers





generators

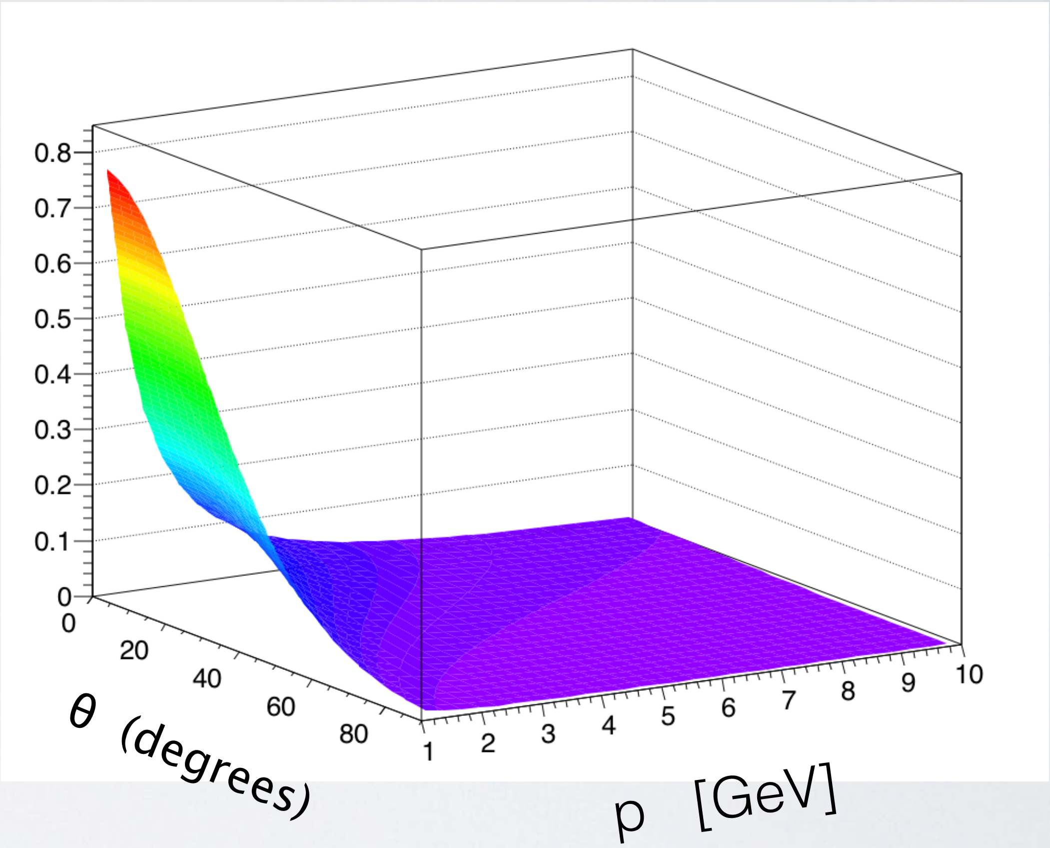

Internal: up to 3 particles
One primary two “beam”
Lund, BEAGLE Format (text)
SLAC Formats
(StdHep, lXDR)
Modello raggi cosmici
GEMC
Easy to add others

esempio di modello raggi cosmici

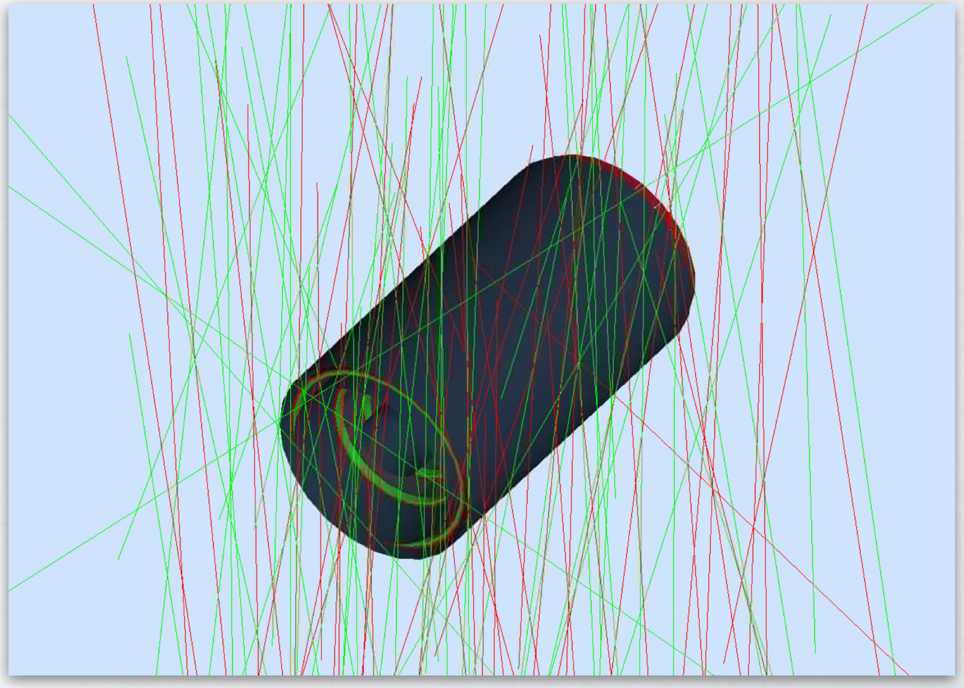
$$\frac{a^{cos(b*\theta)}}{p^2}$$

a =55.6
b=1.04
c=64
p >1 GeV

A. Dar, Phys.Rev.Lett, 51,3,p.227 (1983)



GEMC:
define parameters (or use "default"), p range
define target "AREA"
(x, y, z), R



Output

```
> BST  100, 0

  > True Step by Step infos  (101, 0)
    - Edep                   (101, 1)
    - Pid                    (101, 2)
    - positions              (101, 3)

  > Dgtz Step by Step infos  (102, 0)
    - ADCL                   (102, 1)
    - ADCR                   (102, 2)

  > True Integrated infos    (103, 0)
    - Edep                   (103, 1)
    - Pid                    (103, 2)
    - positions              (103, 3)

  > Dgtz Integrated infos    (104, 0)
    - ADCL                   (104, 1)
    - ADCR                   (104, 2)

  > Voltage as a function of time (105, 0)
    - Identifier                  (105, 1)
    - Time                        (105, 2)
    - Voltage                     (105, 3)

 > Trigger Bank              (106, 0)
    - Identifier             (106, 1)
    - Time                   (106, 2)
    - Voltage                (106, 3)
```

```
"pid",     "Ri", "ID of the first particle");
"mpid",    "Ri", "ID of the mother of the first particle entering the volume");
"tid",     "Ri", "Track ID of the first particle entering the volume");
"mtid",    "Ri", "Track ID of the mother of the first particle entering the volume");
"otid",    "Ri", "Track ID of the original track");
"trackE",  "Rd", "Energy of the track");
"totEdep", "Rd", "Total Energy Deposited");
"avg_x",   "Rd", "Average X position in global reference system");
"avg_y",   "Rd", "Average Y position in global reference system");
"avg_z",   "Rd", "Average Z position in global reference system");
"avg_lx",  "Rd", "Average X position in local reference system");
"avg_ly",  "Rd", "Average Y position in local reference system");
"avg_lz",  "Rd", "Average Z position in local reference system");
"px",      "Rd", "x component of momentum of the particle");
"py",      "Rd", "y component of momentum of the particle");
"pz",      "Rd", "z component of momentum of the particle");
“vx",      "Rd", "x component of primary vertex of the particle");
"vy",      "Rd", "y component of primary vertex of the particle");
"vz",      "Rd", "z component of primary vertex of the particle");
"mvx",     "Rd", "x component of primary vertex of the mother of the particle");
"mvy",     "Rd", "y component of primary vertex of the mother of the particle");
"mvz",     "Rd", "z component of primary vertex of the mother of the particle");
"avg_t",   "Rd", "Average time");
"hitn",    "Ri", "Hit Number");
```

sector
SuperLayer
Layer
wire
LR
Doca
SDoca
time
Stime

Modelli fisici

-PHYSICS="HADRONIC + <EM> + <HP> + <OPTICAL>"

Hadronic can be:

- CHIPS
- FTFP_BERT
- FTFP_BERT_TRV
- FTFP_BERT_HP
- FTF_BIC
- LHEP
- QGSC_BERT
- QGSP
- QGSP_BERT
- QGSP_BERT_CHIPS
- QGSP_BERT_HP
- QGSP_BIC
- QGSP_BIC_HP
- QGSP_FTFP_BERT
- QGS_BIC
- QGSP_INCLXX

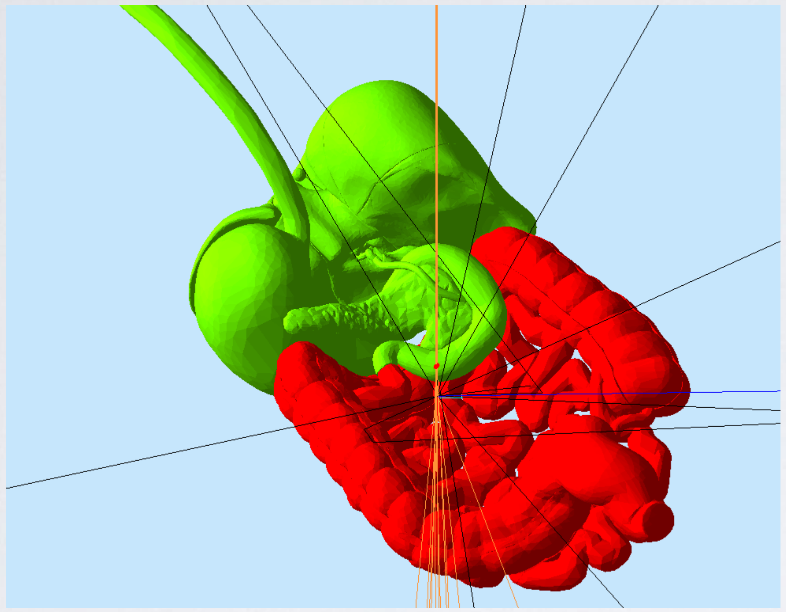
EM can be

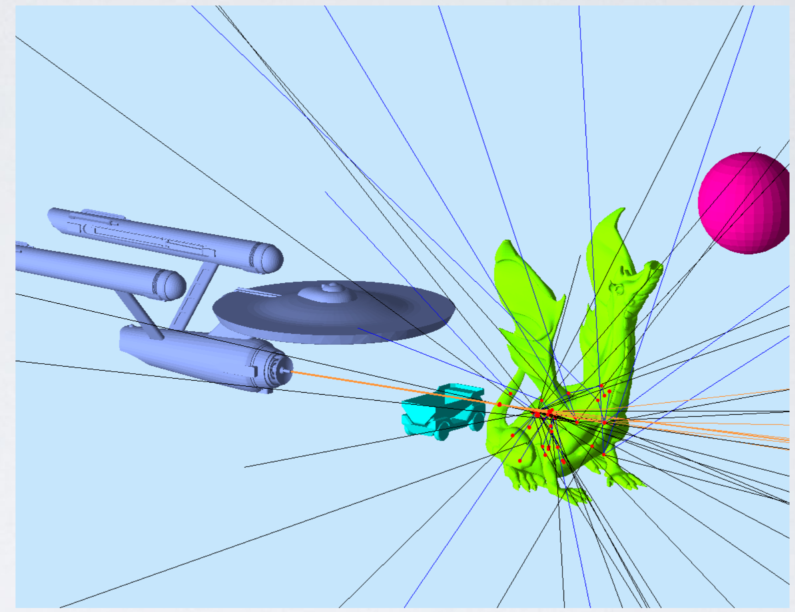
- STD
- EMV
- EMX
- EMY
- EMZ
- LIV
- PEN

HP: High Precision cross sections (e.g. thermal neutron, very low energy processes, etc)

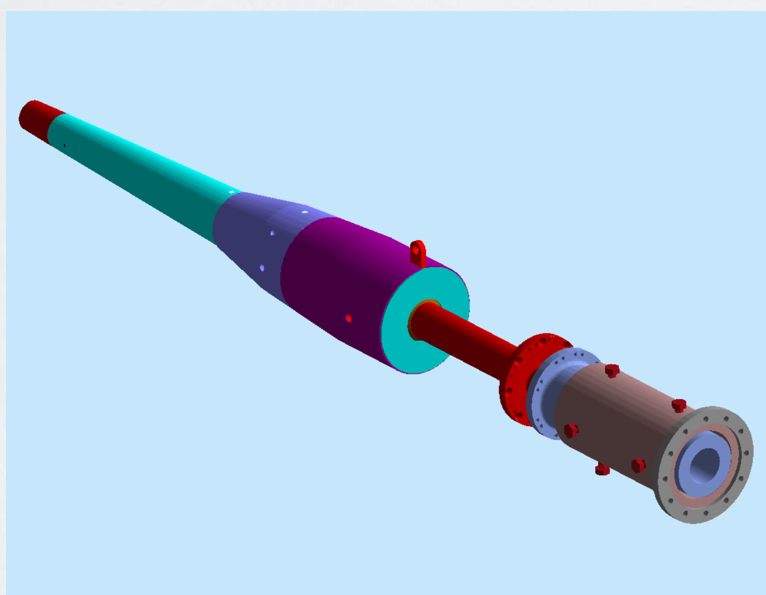
Optical: Activate optical processes

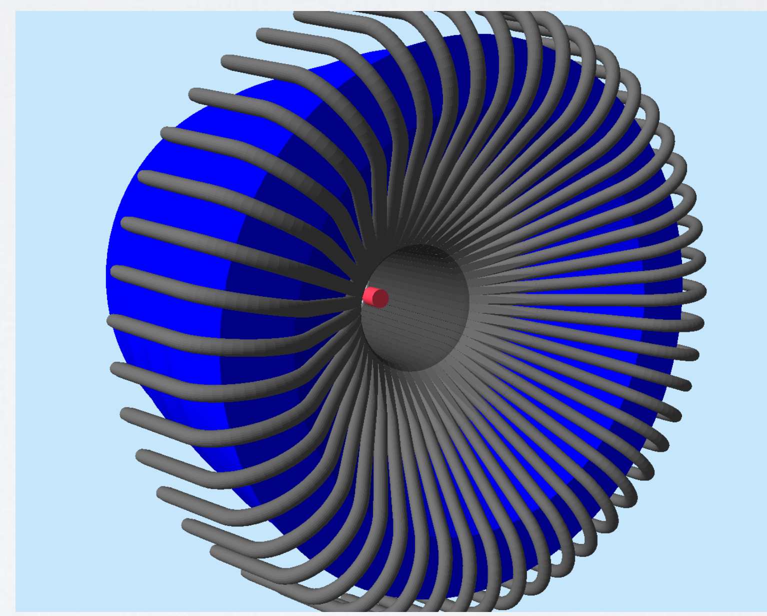
Modelli CAD / GDML / STL





CLAS12 Beamline: CAD engineering drawings

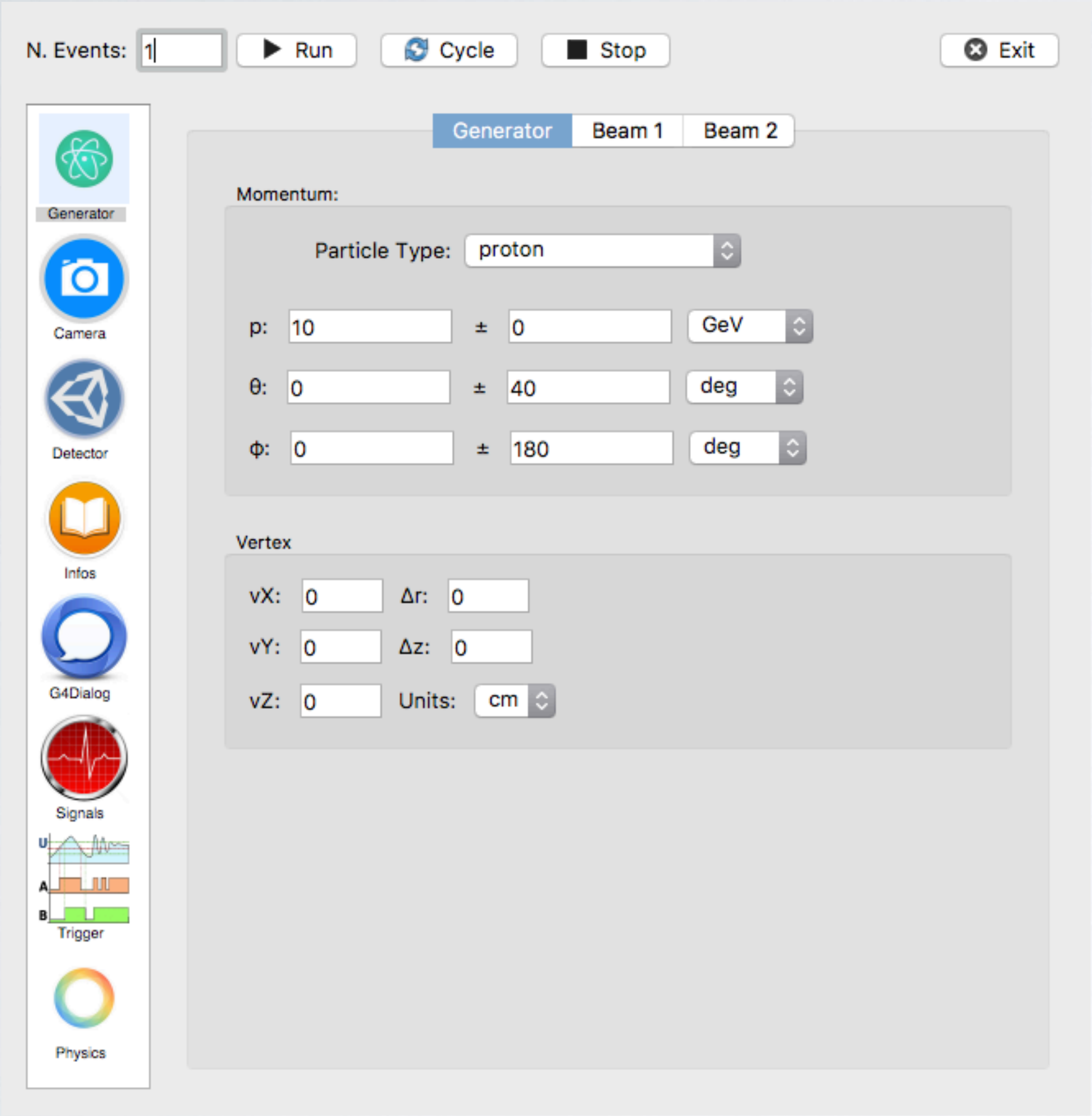


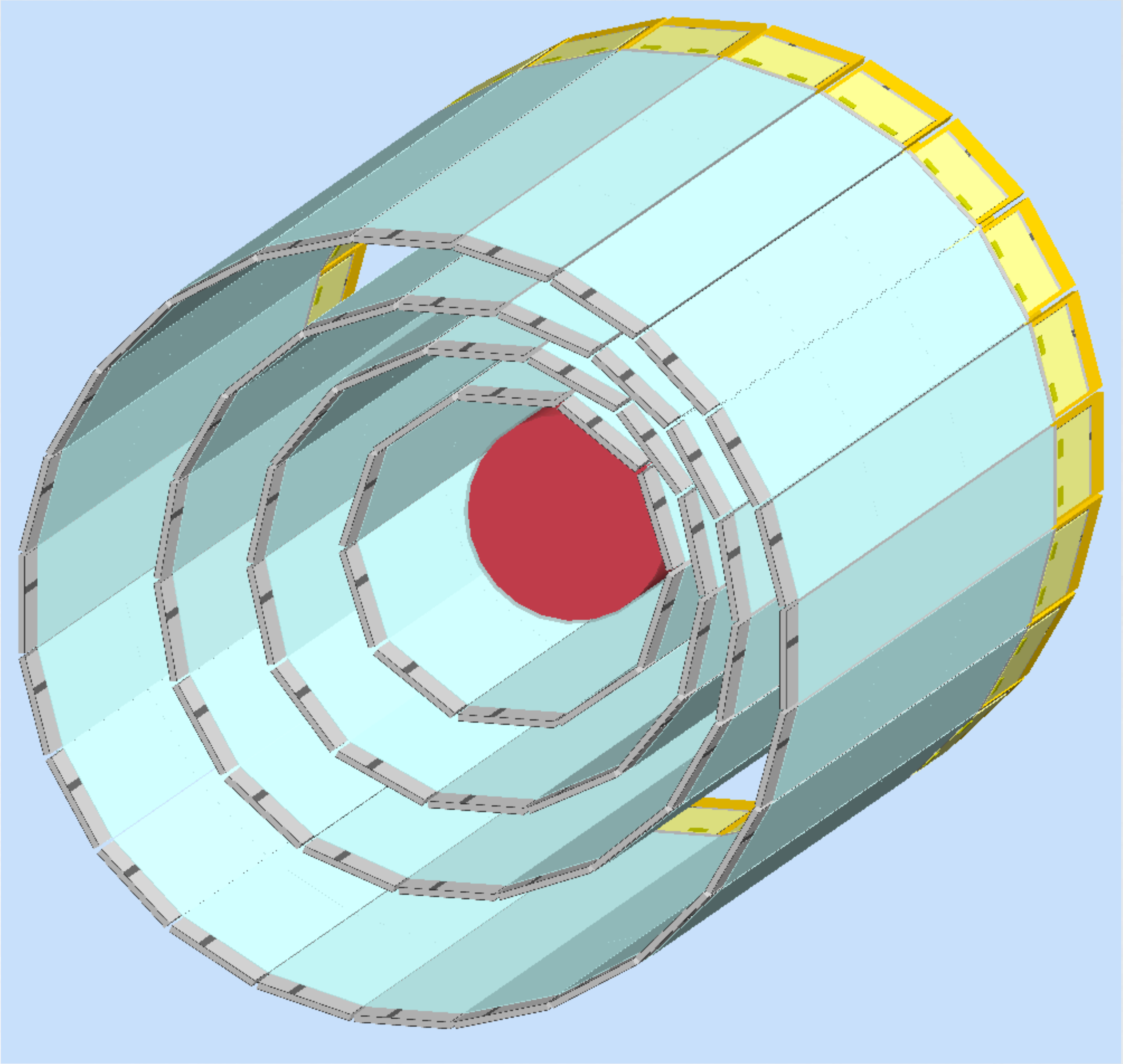


CLAS12 CTOF and light guides: CAD engineering drawings

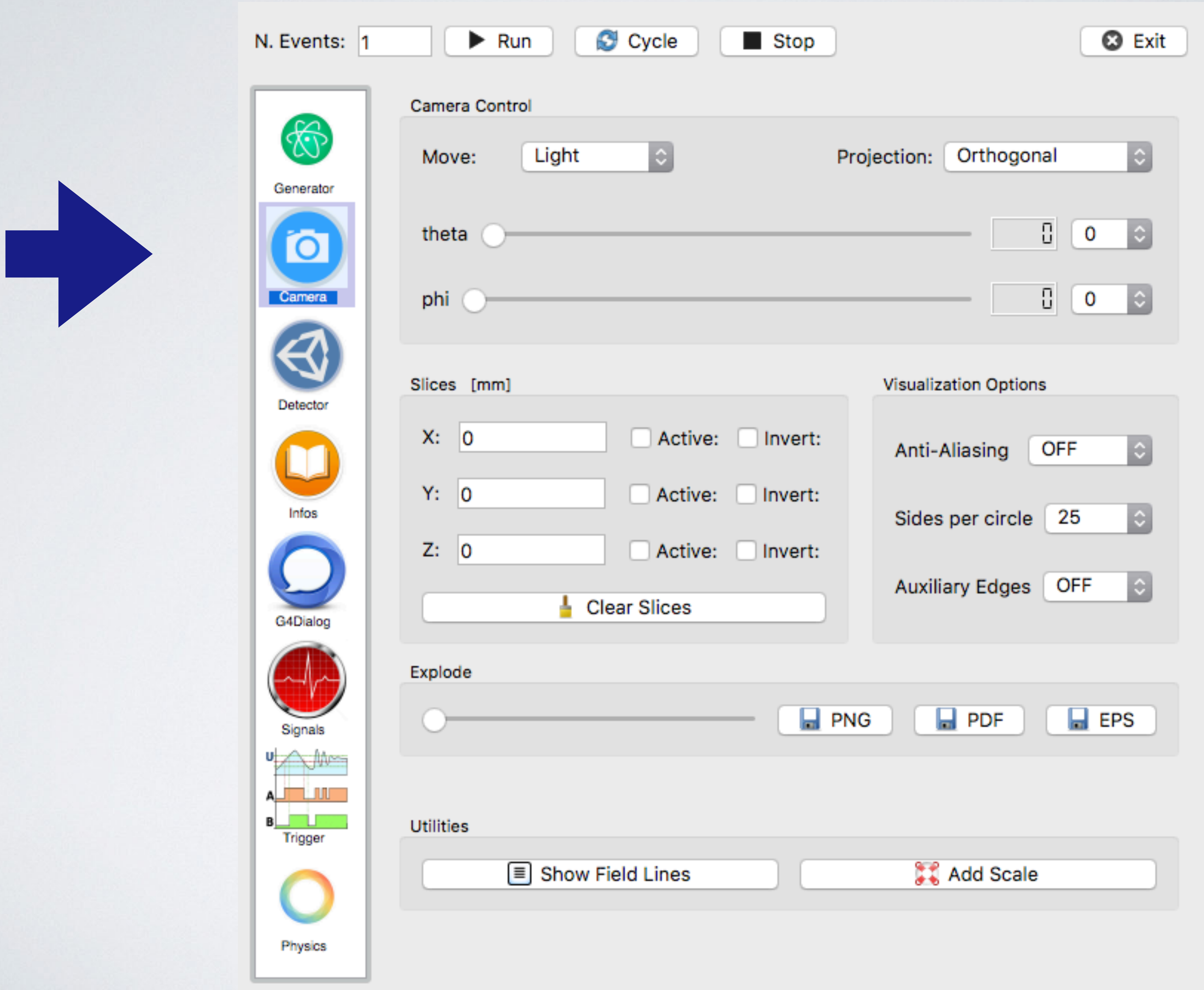
- CAD: objects can be made sensitive at run time.
- Attributes (material, mother volume, position, rotation, touchable ID) can be assigned at run time.
- Mix and match of several factories: TEXT, GDML, CAD

Interfaccia Grafica

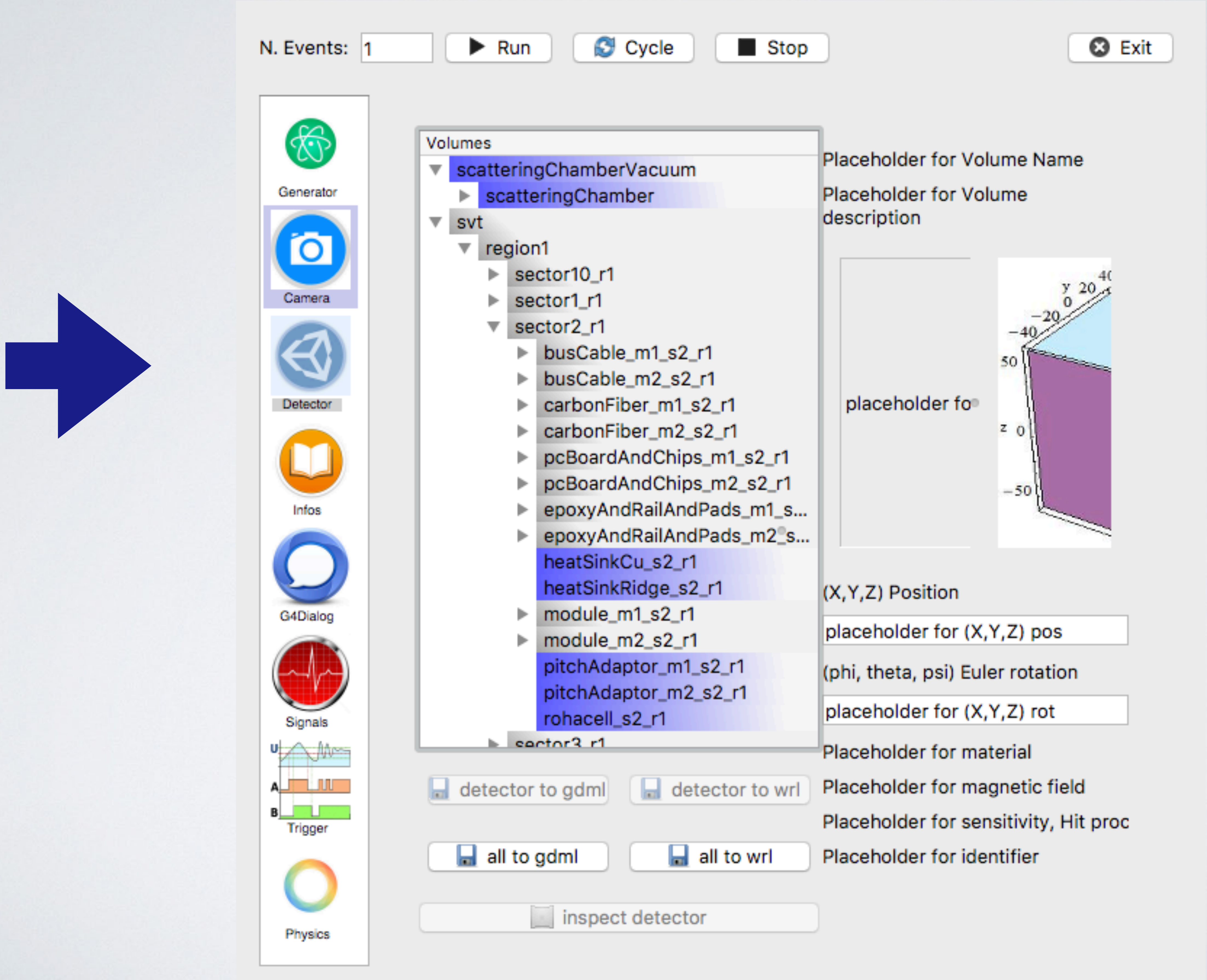




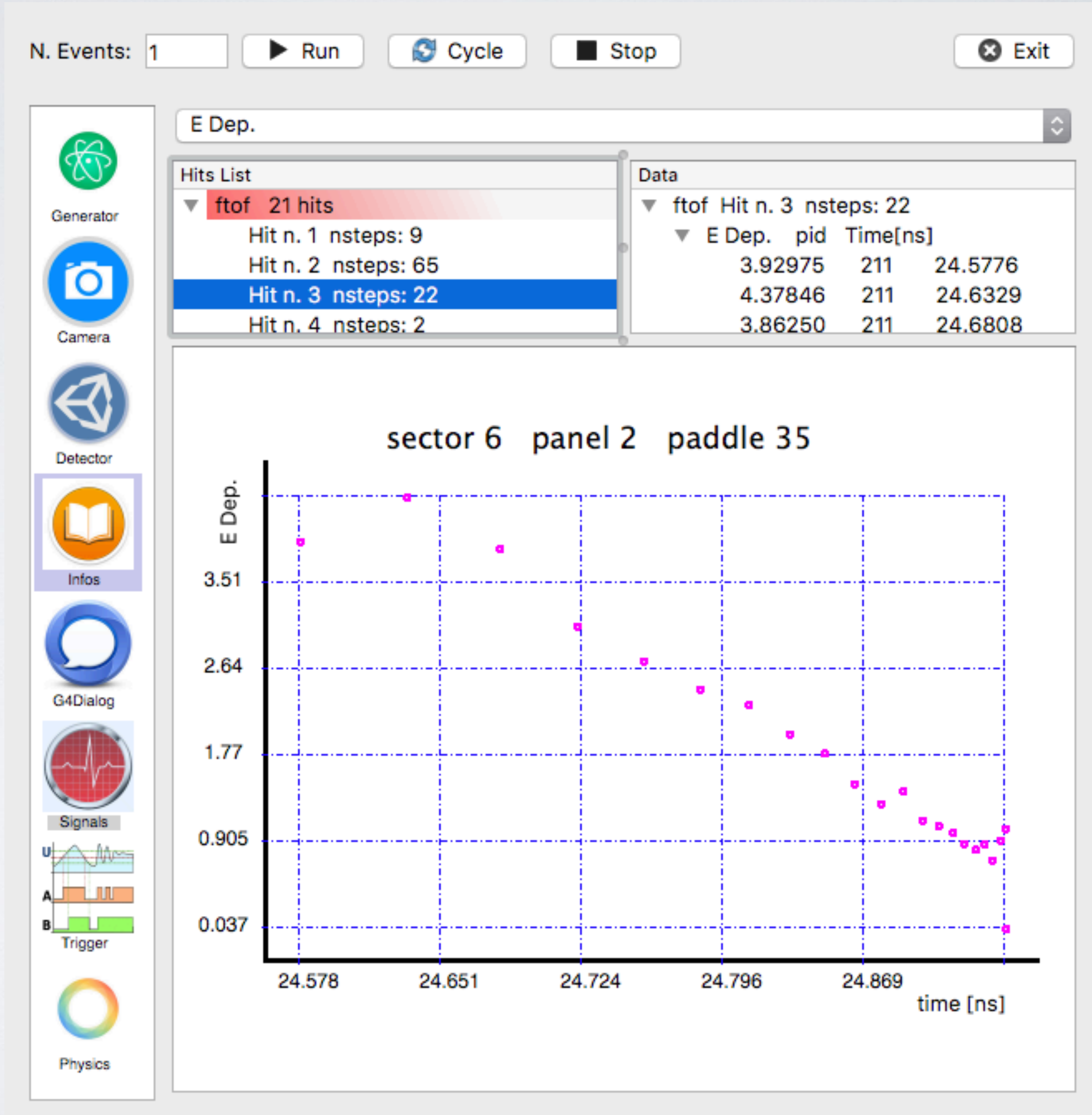
Interfaccia Grafica



Interfaccia Grafica



Interfaccia Grafica



downloads, documentation, examples:
(avviso: stiamo passando da perl a python)

gemc.jlab.org
github.com/gemc

Any questions: ungaro@jlab.org

Come contribuire allo sviluppo:

Create un git "issue".
Oppure:

1. Fork
2. Modify
3. Pull request