

# 粒子物理与核物理实验中的数据分析

---

杨振伟  
清华大学

第六讲:Geant4 的探测器模拟介绍(1)

# 本讲要点

## ■ Geant4 简介

- 一般介绍和历史
- 基本概念和内核
- 重要的用户类

## ■ 如何安装Geant4

## ■ 如何使用Geant4

(探测器几何与材料)

**Geant 4**  
 Geant4 is a toolkit for the simulation of the passage of particles through matter. It has been developed and maintained by a world-wide Collaboration of approximately 100 scientists.

ATLAS at LHC CERN

Its application areas include high energy physics, astrophysics and nuclear physics experiments, medical, accelerator and space science studies.

Borexino at Gran Sasso Laboratory

ESA XMM X-ray telescope

CMS at LHC CERN

BaBar at SLAC

High energy  $\mu$  Courtesy of L3

Photon attenuation. Low energy photons Courtesy of the Italian Nat. Inst. for Cancer Research

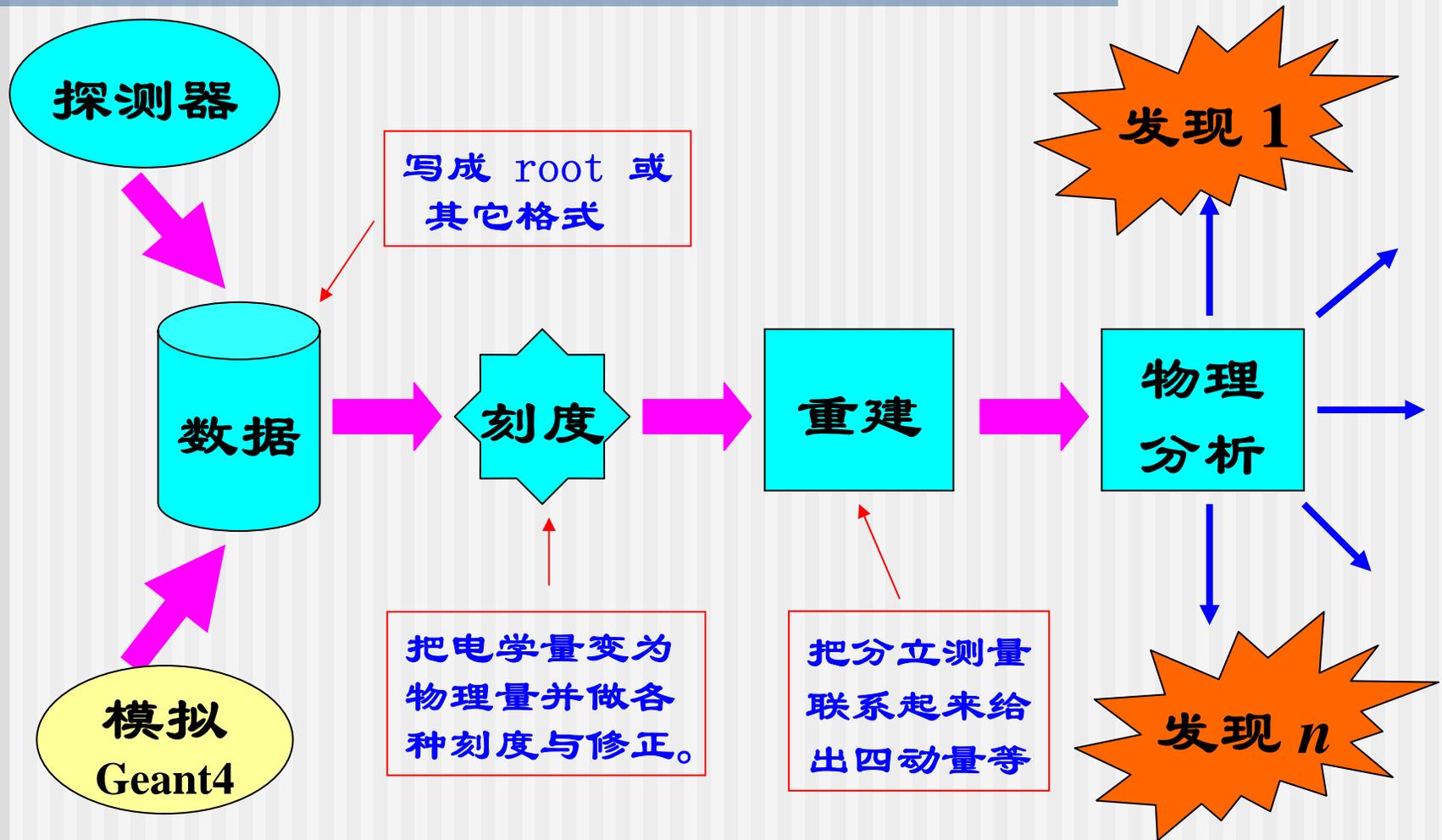
Neutrons Courtesy of CMS

Stopping  $\alpha$  absorption nuclear deexcitation

Geant4 exploits advanced Software Engineering techniques and Object Oriented technology to achieve transparency of physics implementation.

Budker Inst. of Physics IHEP Protivko MEFHI Moscow Pittsburg University

# 数据分析为什么需要模拟信息？



问题在于我们如何知道所做的工作是正确的？

# 模拟需要做的主要工作

- **定义探测器几何** (具有有限接受范围的探测器)
- **指定探测器的物质** (“有用的”/“没用的”各种材料)
- **指定物理过程与粒子** (什么东西在探测器中干了什么)
- **产生原始事例** (模拟重复实验n次)
- **数据收集和分析** (取出探测器可以获得的信息并处理分析)

除了最后的分析，其余都可以由Geant4负责处理  
最后一步一般由ROOT处理

# 什么是 Geant4?

- 简单地说，**Geant4** 是**模拟粒子穿过物质**的工具包。提供了探测器模拟的完整工具：几何，探测器响应，运行、事例以及径迹管理、图形显示、用户接口等。提供了极为丰富的可供选择的物理过程  
详见<http://geant4.cern.ch/support/about.shtml>
  - 其应用领域主要包括：**高能物理、核物理、加速器物理以及医学和空间科学的研究等。**
  - 参考资料
    - 1) <http://geant4.cern.ch>
    - 2) *Nuclear Instruments and Methods in Physics Research A* [506 \(2003\) 250-303](#), and *IEEE Transactions on Nuclear Science* [53 No. 1 \(2006\) 270-278](#).
- 最新版为9.2.patch01，于2009年3月16日发布

# Geant4的特点?

- Geant4 发展于 Geant3, 充分利用了C++语言中的优势, 是粒子与核物理实验模拟最好的工具之一。  
**免费下载, 更新迅速**
- Geant4 是最早成功使用面向对象环境重新设计的粒子与核物理软件包, 几乎所有大型实验的模拟都基于Geant4。
- Geant4 的应用远不止于粒子与核物理。考虑了各个方面的需求, 包括重离子物理、电荷与宇称破缺物理、宇宙线物理、天体物理、空间科学以及医学应用等。
- 为了满足这些需求, 提供了强大的功能和灵活性。

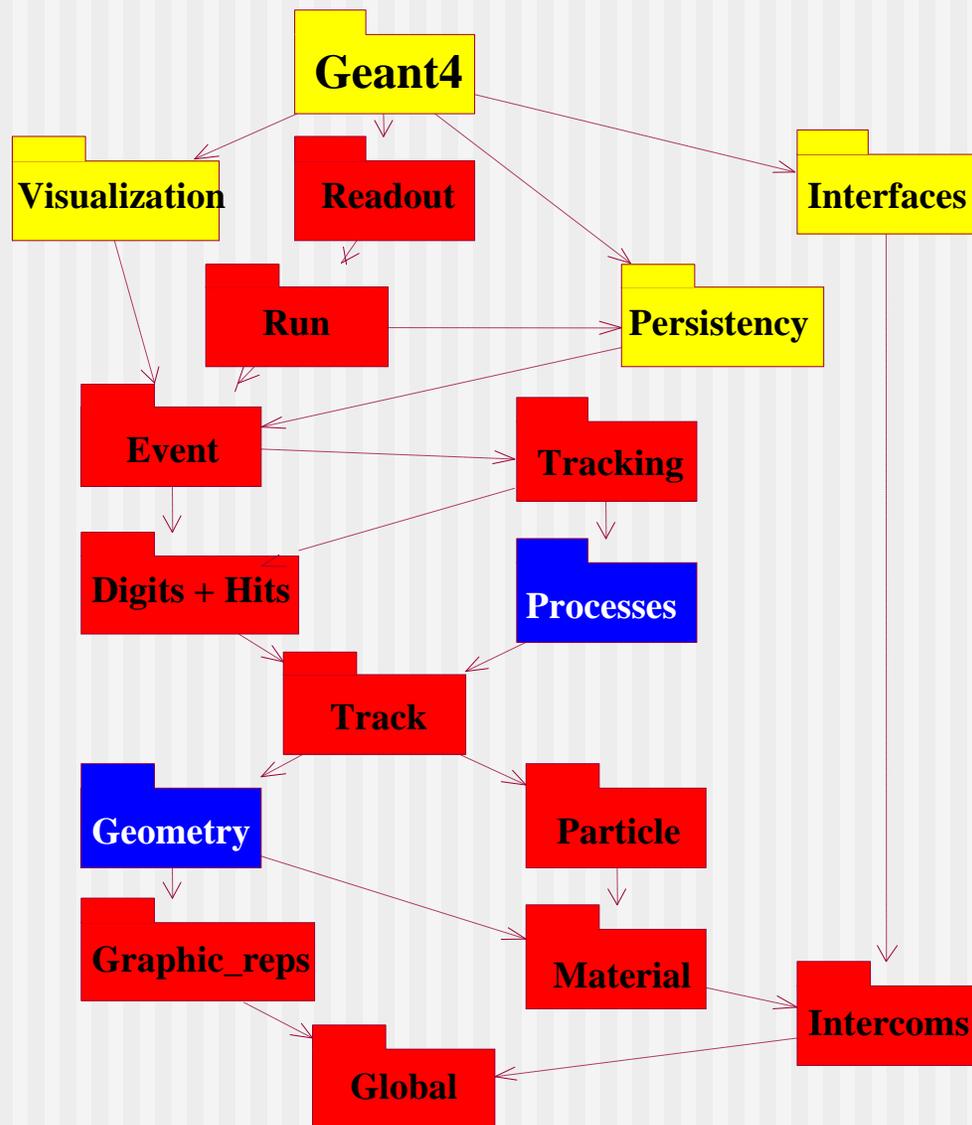
# Geant4 的基本概念与内核

## ■ Geant4 包含17大类

- 每一大类都有专门独立的研发工作小组负责。
- 各大类之间的界面 (例如顶层的设计) 由整体结构工作小组来维护。

## ■ Geant4 的内核

- 处理运行号, 事例, 迹, 步, 击中, 运动轨迹。
- 提供几何表示和物理过程的架构。



# Geant4 软件包中的内容

## ■ 代码

- ~1 百万行语句
- 网上可以自由下载
- 还在不断的增加与更新

## ■ 相关文件说明

- 6 份使用工具书
- 网上可以自由下载

## ■ 应用例子

- 与软件包一起附送
- 参照工具书运行例子程序
- 包含了从简单到较复杂各种模拟程序  
(novice, extended, advanced)

## ■ 平台

- Linux, MacOSX, SUN
- Windows/XP

## ■ 需要预安装的软件

- C++ 编译器
- 如g++, VC++等
- CLHEP库
- GNU Make, shell

## ■ 图像与用户界面

- X Windows
- OpenGL 或 MesaGL
- DAWN
- ...

至少需要其中的一个

# 安装Geant4(概要)

对SLC3/4操作系统的支持和兼容性很好，如果SLC3/4是完全安装，则安装Geant4只需要预安装CLHEP(必需)和Qt4(可选，Geant4.9.0开始尝试支持Qt4，但还不完善)

- 1) 下载并安装CLHEP程序包
- 2) 如果需要Qt4，则下载并安装Qt4
- 3) 下载Geant4软件包以及相应的数据文件(用于各种物理过程模型)，按照安装手册进行编译安装

安装过程中，"./Configure"时有很多选项需要选择，根据自己的需要选择。

参见

[http://hep.tsinghua.edu.cn/~yangzw/software/install\\_clhep.txt](http://hep.tsinghua.edu.cn/~yangzw/software/install_clhep.txt)

[http://hep.tsinghua.edu.cn/~yangzw/software/install\\_qt4.txt](http://hep.tsinghua.edu.cn/~yangzw/software/install_qt4.txt)

[http://hep.tsinghua.edu.cn/~yangzw/software/install\\_geant4.txt](http://hep.tsinghua.edu.cn/~yangzw/software/install_geant4.txt)

# 安装Geant4(概要) 续

对于其它Linux发行版，安装过程类似。

Ubuntu用户安装Qt4不需要编译，可以直接从安装源进行安装

如果系统版本相同(内核版本和g++版本)，把已经编译好的程序直接复制到其它机器上即可使用。

比如，对SLC3系统，直接复制training服务器

/projects/soft/ext/clhep.tgz和g4.tgz到本地机器，解压缩到相应目录即可。

对于Ubuntu用户，参照

[http://hep.tsinghua.edu.cn/~yangzw/software/ubuntu\\_soft/Readme.txt](http://hep.tsinghua.edu.cn/~yangzw/software/ubuntu_soft/Readme.txt)

# 设置Geant4环境变量

由于Geant4练习过程中有大量图形显示，希望大家有条件的话在本地机器上进行练习。本地机器Geant4安装在  
`/projects/soft/ext/g4/geant4.9.0.p01/`

运行Geant4之前需要先设置Geant4的环境变量，即运行：  
`source /projects/soft/ext/g4/geant4.9.0.p01/env.sh`

Geant4环境变量都以G4开头。比如

G4INSTALL : 安装目录

G4SYSTEM : 操作系统及c++编译器名称,如Linux-g++

G4WORKDIR : 工作目录

我们设定的工作目录为`/projects/$USER/geant4`

所以大家需要在`/projects/$USER`目录下建立geant4子目录。

以后所有的geant4工作都在G4WORKDIR下进行。

# 例一： Novice Example N01

Geant4自带很多例子，从新手入门、高级、专家级例子：  
\$G4INSTALL/examples/novice目录为入门级7个例子

首先看最简单的例子

```
cd $G4WORKDIR
```

```
cp -a $G4INSTALL/examples/novice .
```

```
cd novice/N01
```

```
gmake          (编译)
```

```
exampleN01     (执行)
```

这是最简单的例子，只是告诉你Geant4如何工作，并且确认你安装的Geant4可以工作。

例子中定义了径迹探测器和三明治式量能器，每个事例发射一个虚拟粒子geantino(没有真实的物理作用)。

后面我们会具体介绍如何定义探测器几何与材料、如何定义入射粒子、如何添加需要的物理过程、如何记录需要的实验数据。

# 最基本的主函数(exN01)

```
#include "G4RunManager.hh"  
#include "G4UImanager.hh"  
#include "ExN01DetectorConstruction.hh"  
#include "ExN01PhysicsList.hh"  
#include "ExN01PrimaryGeneratorAction.hh"
```

```
int main()  
{
```

```
    //构造默认的运行管理器(强制)
```

```
    G4RunManager* runManager = new G4RunManager;
```

```
    //用探测器以及物理过程初始化运行管理器(强制)
```

```
    G4VUserDetectorConstruction* detector = new ExN01DetectorConstruction;
```

```
    runManager->SetUserInitialization(detector);
```

```
    G4VUserPhysicsList* physics = new ExN01PhysicsList;
```

```
    runManager->SetUserInitialization(physics);
```

```
    // 设置用户行为(强制)
```

```
    G4VUserPrimaryGeneratorAction* gen_action = new ExN01PrimaryGeneratorAction;
```

```
    runManager->SetUserAction(gen_action);
```

```
    //.....接下页
```

探测器、物理过程以及用户行为的定义随模拟的问题而变，在相应类中指定。

# 最基本的主函数(exN01)

```
//接上页...
// 初始化G4 内核
runManager->Initialize();

// 获取UI管理器的指针，并设置verboseities。
G4UImanager* UI = G4UImanager::GetUIpointer();
UI->ApplyCommand("/run/verbose 1");
UI->ApplyCommand("/event/verbose 1");
UI->ApplyCommand("/tracking/verbose 1");

// 开始一次运行
G4int numberOfEvent = 3;
runManager->BeamOn(numberOfEvent);

// 结束作业
// 释放内存：用户行为、物理过程以及探测器描述属于运行管理器，
// 将被运行管理器自动删除，所以不应该在主函数中删除之。
// 只需要删除运行管理器和其它动态指针即可。
delete runManager;

return 0;
}
```

如果想图形显示探测器以及物理过程的信息，  
需要在main函数中用到G4VisManager

# 添加如下代码进行图形显示

```
G4VisManager* visManager = new G4VisExecutive;  
visManager->Initialize();  
UI->ApplyCommand("/control/execute vis.mac");  
//vis.mac文件中定义了跟图形显示有关的一些命令，比如观看角度、缩放比例等。
```

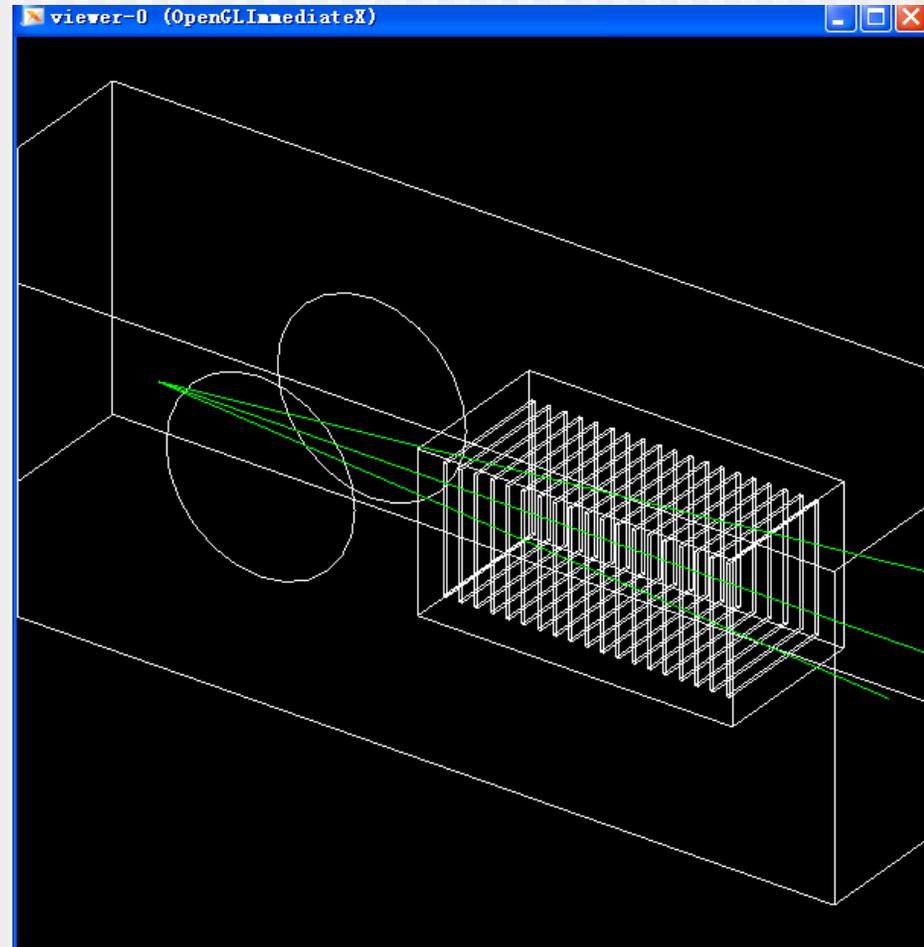
为了让事例显示在程序运行后仍然驻留，  
需要开启一个G4界面(G4UIsession)

```
G4UIsession * session = 0;  
session = new G4UITerminal();  
session->SessionStart();
```

注1：确保已经include了头文件  
**G4VisExecutive.hh**

注2：确保已定义了**G4VIS\_USE**环境变量

注3：程序结束前，不要忘记删除**session**  
和**visManager**两个动态指针。



# 如何定义探测器几何和材料

Geant4模拟首先要考虑的是模拟的物理问题主要包括哪些物理过程以及想采用的探测器构造。例N01的探测器是在ExN01DetectorConstruction类中定义的，该类继承于G4VUserDetectorConstruction。

探测器构造主要包括：几何、材料、灵敏区域、灵敏区域的读出机制，本讲只涉及几何与材料。详见该类的定义以及实现，即ExN01DetectorConstruction.hh(cc)

**关键：** 在Construct()函数中构造必要的物质、必要的几何形状，以及如何将基本单元放置(架构)成探测器。还可能包括磁场、灵敏区域定义等。

# 如何定义探测器材料(1)

物质定义:

isotopes <-> G4Isotope  
elements <-> G4Element  
molecules, compounds, mixtures <-> G4Material

G4Material属性包括温度、压强、状态、密度等，定义时有时需要明显指出属性。

```
double density = 1.390*g/cm3;//密度  
double a = 39.95*g/mole; //原子量  
double z=18.; //原子序数  
G4Material* lAr = new G4Material("liquidArgon",z,a,density);
```

```
G4Element* H = new G4Element(name="Hydrogen",symbol="H" , z= 1., a);  
G4Element* O = new G4Element(name="Oxygen" ,symbol="O" , z= 8., a);  
density = 1.000*g/cm3;  
G4Material* H2O = new G4Material(name="Water", density,  
ncomponents=2);  
H2O->AddElement(H, natoms=2);  
H2O->AddElement(O, natoms=1); //定义水，给定密度、元素种类数目、添加元素
```

注: Geant4中的数据如果有单位都要明显给出单位。

比如double length = 1.0\*cm

= 10.0\*mm

二者效果是一样的。

方便程序中使用合适的单位，而且看代码的时候比较直观。

# 如何定义探测器材料(2)

## //空气的定义

```
G4Element* N = new G4Element("Nitrogen", "N", z=7., a= 14.01*g/mole);
G4Element* O = new G4Element("Oxygen" , "O", z=8., a= 16.00*g/mole);

G4Material* Air = new G4Material("Air", density= 1.29*mg/cm3, nel=2);
Air->AddElement(N, 70*perCent);
Air->AddElement(O, 30*perCent); //指定百分比
```

## //定义真空

```
G4double a, z;
G4double density, temperature, pressure;
G4int nel;
G4Material* Vacuum =
new G4Material("Galactic", z=1., a=1.01*g/mole,
              density= universe_mean_density,
              kStateGas, 2.73*kelvin, 3.e-18*pascal); //指定温度、压强等属性
```

# 如何定义探测器几何(1)

探测器几何在Geant4中是由很多“**体积**”(Volumes)构成的。Volume的概念在几何构造中非常重要。

- **三个概念层面**
  - **G4VSolid** – **形状, 尺寸**
  - **G4LogicalVolume** – **子物理体积, 材料, 灵敏区, 磁场, 用户限制等等。**
  - **G4VPhysicalVolume** – **位置, 转动**
- **原则上几何描述的三个层面允许最大限度的信息再利用, 以便减小内存空间。**
- **探测器灵敏区应由用户在**G4VSensitiveDetector**进行具体的定义并设置到**G4LogicalVolume**上。**

# 如何定义探测器几何(2)

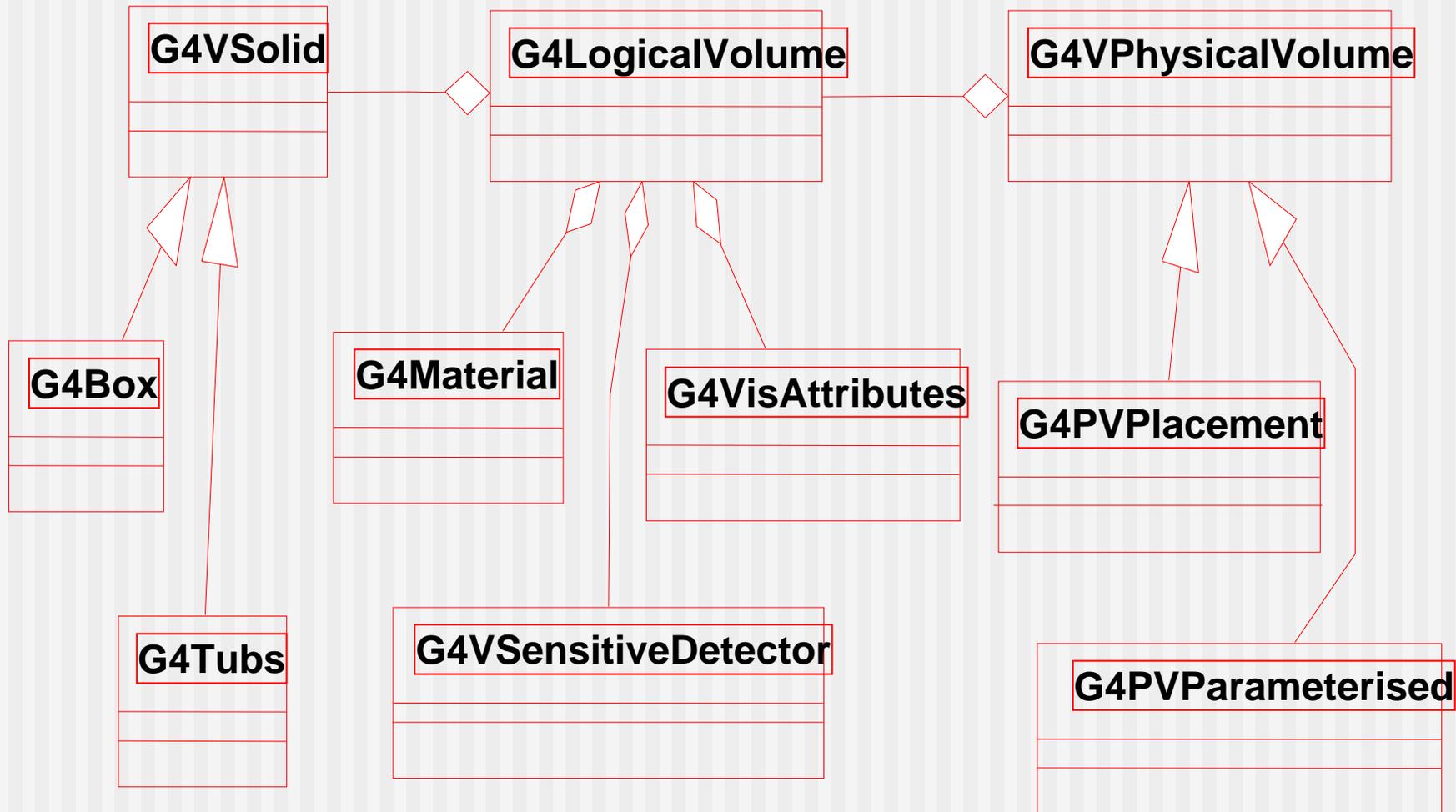
定义一个G4Box，长方体

```
G4double expHall_x = 3.0*m;  
G4double expHall_y = 1.0*m;  
G4double expHall_z = 1.0*m;  
//Solid, 指定几何形状和尺寸  
G4Box* experimentalHall_box  
    = new G4Box("expHall_box",expHall_x,expHall_y,expHall_z);  
//Logical, 指定具体物理特性, 如其中物质为Ar气  
experimentalHall_log = new G4LogicalVolume(experimentalHall_box,  
                                             Ar,"expHall_log",0,0,0);  
//Physical, 指定放置位置以及旋转角度等  
experimentalHall_phys = new G4PVPlacement(0,G4ThreeVector(),  
                                           experimentalHall_log,"expHall",0,false,0);
```

Geant4中基本的几何形状很多，比如：

G4Box, G4Tubs, G4Cons, G4Para, G4Trd, G4Trap,  
G4Spere, G4Torus, G4Polycone,...

# 一个 Geant4 体积的调用示意图



# 作为用户，如何使用Geant4？

- **首先不要尝试去读懂 Geant4 上百万行语句的含义！**
- **其次不要尝试去读懂所有 Geant4 的例子！**
- **然后根据你的实际情况选择相应的例子，我个人建议大家仔细读懂“Novice”中的第二个例子N02中在 `src` 与 `include` 中的每一行语句的含义。**
- **Geant 4 需要用户告诉探测器的几何是什么？材料是什么？研究的粒子是什么？物理过程什么？**
- **用户需要从 Geant4 中得到粒子在反应后或者输运后，可以被探测器记录的击中数，末态粒子数，对应的四动量，能损大小等等。然后把所有物理量都转化为电学量，如光电子数，电子学道数等等。**

# 小结

---

- **模拟探测器的必要性**
- **Geant4 模拟软件包的基本概念**
- **安装Geant4和设置环境变量**
- **模拟探测器的简单例子**
- **图形显示**
- **探测器几何和材料的定义**

# 练习

1. 修改稿例子N01，加上图形显示。  
从不同视角查看探测器显示情况。  
(hints: 将N02中的vis.mac复制到N01目录，  
利用其中关于vis的UICommand命令。)
2. 修改N01中探测器的物质以及几何形状，在  
tracker tube的内部中心平行放置一个内径  
20cm,外径25cm，长度40cm的小圆管，管的  
材质为硅。
3. 编译运行N02例子，查看探测器以及物质构造。
4. 阅读Geant4 User's Guide (for Application  
Developers)第2.1-2.3节，4.1-4.3节。
5. 下载Geant4.9.2.p01源码，尝试安装Geant4

# 参考资料

---

1. <http://geant4.cern.ch>
2. Geant4 User's Guide (for Application Developers)第2.1-2.3节, 4.1-4.3节。  
可在  
<http://hep.tsinghua.edu.cn/~yangzw/CourseDataAna>目录下载
3. Geant4安装手册
4. NIM [A 506 \(2003\) 250-303](#), and  
*IEEE Transactions on Nuclear Science* [53 No. 1 \(2006\) 270-278](#).