Analysis Tools And Managers Classes

Guy Barrand, LAL
I. Hrivnacova, IPN Orsay

16th Geant4 Collaboration Meeting, 19 - 23 September, SLAC

Outline

- G4tools (from Guy)
  - ioda
  - inlib & exlib
  - g4tools
- Analysis managers
  - Why manager classes in Geant4
  - Current implementation (and limitations)
  - Example of usage

g4tools

ioda -> inlib/exlib -> g4tools

iPad & Android -> ioda

- 2010 : iPad, Android tablets : we can’t ignore that.
- A new way to distribute world wide applications! We can’t ignore that too!
- ioda : visualization of HEP and astro data at various file formats targeted for iDevices and Android devices: http://ioda.lal.in2p3.fr
- A « 1.x » on the AppStore and Android market since begin of 2011. Available also on desktops (UNIXes, Windows).
- (I can show easily to my kids what I do at LAL! ☺)

ioda -> inlib & exlib

- C, GL-ES is a common basement
- Android NDK and iOS permit C++. (Despite that the promoted environments are java and Objective-C based).
- Then we continue with C++ as our primary OO language.
- GL-ES induces that we have to get rid of our Inventor/coin3d historical choice for the graphics.
- Need new software (than the OpenScientist choices).
- Occasion to rethink a little bit things…

inlib & exlib

- inlib: C++ code over STD, STL and system functions considered as defacto standards (as socket ones).
- exlib: C++ code over inlib and « various other external packages » as GL-ES, zlib, expat, freetype, jpeg, png, etc…, external packages coming with or easily installable on iOS, Android, UNIXes, Windows.
- all code in inlib/exlib is inlined and « pure header code »!

inlib & exlib

- This permits to use for application as ioda, the native IDEs as Xcode, Android NDK make system. (Having no lib to manage, working with the native IDEs is easy: only some –I to declare). (Native IDEs are the simplest ways to build packing for the « markets »).
- It simplifies a lot… everything!
- In particular the distribution: since « pure headers », no need for a « binary installation pass » for the user. It bypasses de facto a « forever painful » point of user support: a huge relief.
- “g++ -g” for development, “g++ -O” when distributing. It is workable.

Histos, ntuples, IO

- inlib contains histogram classes. (An inlining of the code of OpenScientist/HCL package).
- It contains code to write histograms and “flat ntuples” at the root format! But also at the XML AIDA format and CSV for ntuples.
- In exlib, there is also code over CERNLIB to handle HBOOK histos and ntuples (obviously not available for iOS and Android!)
- These classes are ideal candidates for code to handle “batch analysis tools for G4”.

g4tools

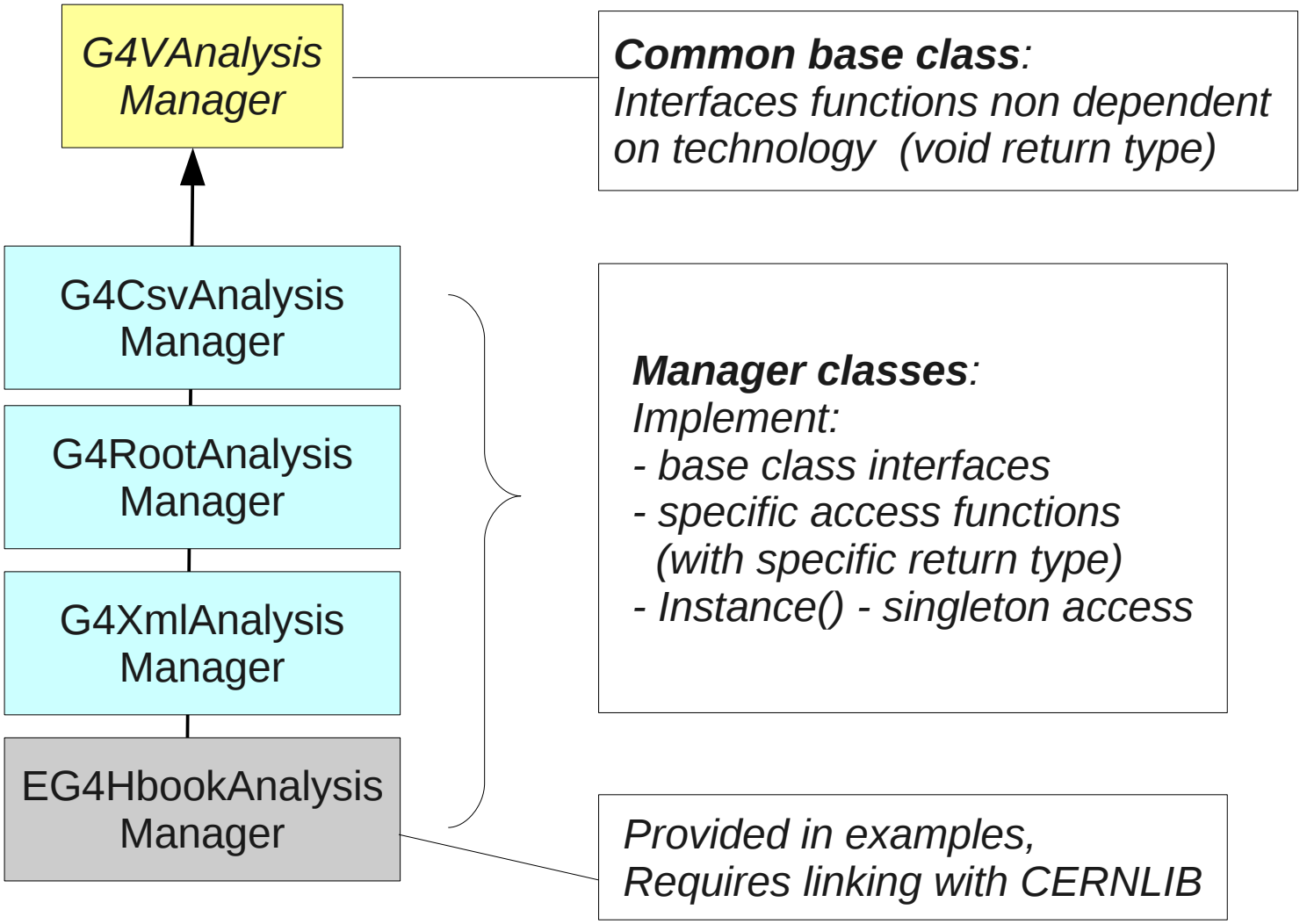
- It is an automatic extraction of inlib/exlib classes for what is needed for a “G4 batch program”.
- Then a program that needs to create/fill histos and ntuples and write them in a file at various formats. A file readable “later” by various analysis tools to do, for example, ntuple projections, fitting and plotting.
- First version available now and able to handle 1D, 2D histos and profiles, flat ntuples, write them in files readable by ioda, paw, root, jas, osc-plot, excel (for ntuple at .csv).

Analysis Manager Classes in Geant4

Why Manager classes?

- Not uniform interfaces in g4tools
  - Differences according to a selected technology (root, XML, HBOOK)
  - Geant4 manager classes hide these differences from the user
- No higher level management of created objects (file, histogram, ntuple) in g4tools
  - Geant4 manager classes provide:
    - Memory management
    - Access to histograms, ntuple columns via indexes

Analysis Managers Implementation



Analysis Category

/geant4/source/analysis

CmakeLists.txt, exception_classification.txt, GNUmakefile, History, **tools.license**

include

g4analysis_defs.hh,
G4RootAnalysisManager.hh
G4XmlAnalysisManager.hh,
G4CsvAnalysisManager.hh
G4VAnalysisManager.hh

tools

args, charmanip, cmp, fmath, hbook, histo, math, mem, mnmx, path, platform, pointer, randf, random, rcmp, realloc, safe_cast, scast, sout, sprintf, srep, sto, strip, stype, tos, typedefs, vdata, version, vfind, vmanip, waxml, wcsv_ntuple, words, wroot

src

G4RootAnalysisManager.cc,
G4XmlAnalysisManager.cc,
G4CsvAnalysisManager.cc,
G4VAnalysisManager.cc

test

README, chbook.cpp, hbook.f, hbooknt.f, hello_f77.f, histo.cpp, ntuple.kumac, rcsv.C, rcsv.kumac, rroot.C, waxml.cpp, wcsv.cpp, whbook.cpp, wroot.cpp
+ build scripts

Histograms

N4RunAction.cc

```
#include "N4Analysis.hh"

void N4RunAction::BeginOfRunAction(const G4Run* run)
{
  // Get analysis manager
  G4AnalysisManager* man = G4AnalysisManager::Instance();

  // Open an output file
  man->OpenFile("exampleN4");

  // Create histogram(s)
  man->CreateH1("0","Edep in absorber", 100, 0., 800*MeV);
  man->CreateH1("1","Edep in gap", 100, 0., 100*MeV);
}

void N4RunAction::EndOfRunAction(const G4Run* aRun)
{
  G4AnalysisManager* man = G4AnalysisManager::Instance();
  man->Write();
  man->CloseFile();
}
```

N4EventAction.cc

```
#include "N4Analysis.hh"

void N4EventAction::EndOfEventAction(const G4Run* aRun)
{
  G4AnalysisManager* man = G4AnalysisManager::Instance();
  man->FillH1(0, fEnergyAbs);
  man->FillH1(1, fEnergyGap);
}
```

N4Analysis.hh

```
#ifndef N4Analysis_h
#define N4Analysis_h 1

#include "g4analysis_defs.hh"

using namespace G4Root;
//using namespace G4Xml;
//using namespace G4Csv;

#endif
```

Selection of the output format at a single place

Ntuple

N4RunAction.cc

```
#include "N4Analysis.hh"

void N4RunAction::BeginOfRunAction(const G4Run* run)
{
  // Get analysis manager
  G4AnalysisManager* man = G4AnalysisManager::Instance();

  // Open an output file
  man->OpenFile("exampleN4");

  // Create ntuple
  man->CreateNtuple("N4", "Edep and TrackL");
  man->CreateNtupleDColumn("Eabs");
  man->CreateNtupleDColumn("Egap");
  man->FinishNtuple();
}
```

N4EventAction.cc

```
#include "N4Analysis.hh"

void N4EventAction::EndOfEventAction(const G4Run* aRun)
{
  G4AnalysisManager* man = G4AnalysisManager::Instance();
  man->FillNtupleDColumn(0, fEnergyAbs);
  man->FillNtupleDColumn(1, fEnergyGap);
  man->AddNtupleRow();
}
```

Analysis Managers Implementation - More (1)

- Specific Managers are singletons:
  - Cannot instantiate two objects of one type, eg. Root manager
  - Two objects of different types can coexist, eg. Csv and Xml
    - Then instead of G4AnalysisManager typedef user has to give a concrete type of each manager:

```
#include "G4CsvManager.hh"
#include "G4XmlManager.hh"

G4CsvManager* csvManager = G4CsvManager::Instance();
G4XmlManager* xmlManager = G4XmlManager::Instance();
```

    - or with an explicit namespace:

```
#include "g4analysis_defs.hh"

G4Csv::G4AnalysisManager* rootManager = G4Csv::G4AnalysisManager::Instance();
G4Xml::G4AnalysisManager* xmlManager = G4Xml::G4AnalysisManager::Instance();
```

Analysis Managers Implementation - More (2)

- Limitations:
  - 1 output file
    - If file extension is not provided in a file name, it is automatically completed according to the file format (.csv, .hbook, .root, .xml)
  - 1 directory for histograms, 1 directory for ntuple
    - Directory names can be changed by the user
  - 1 ntuple
    - With columns of int, float and double type
  - Whatever number of histograms
    - Currently only H1D type interfaced in managers
    - To be extended with H2D, H3D; and P1D, P2D, P3D (profiles)
- The limitations can be reduced following the feedback from users (and G4 developers)

Where to find it

- New analysis category in geant4 SVN
  - trunk/geant4/source/analysis
    - g4tools headers are in source/analysis/include/tools
  - branches/geant4/examples/extended/_symbols/extended_branch/common/analysis
    - HBOOK manager
  - trunk/geant4/source/analysis/test
    - Test programs with direct use of g4tools (without G4 managers)
- First example of use in new novice examples (example N4):
  - branches/geant4/examples/novice/_symbols/novice_branch/N4

Conclusions

- The new analysis tools included in Geant4
- No need for external packages with the code in kernel
  - HBOOK manager requiring CERNLIB is not built with kernel libraries
- Use of g4tools is very simple
- Waiting for a feedback from users before adding more features