IPN
INSTITUT DE PHYSIQUE NUCLÉAIRE
ORSAY

New Developments In Analysis

I. Hrivnacova, IPN Orsay

18th Geant4 Collaboration Meeting,
23 - 27 September 2013, Seville

Outline

- New features
  - Extension for handling more than one ntuple
  - Support for logarithmic binning in 1D, 2D histograms
- Migration to multi-threading
- Class redesign
- Future plans

Handling More Ntuples

- Request for advanced examples
- Extended interface:
  - Existing functions for creating ntuple, ntuple colums and finish ntuple (without ntupleId argument) can be called successively; they are applied to the last created ntuple
  - Existing functions for FillNtupleTColumn() and AddNtupleRow() (without ntupleId argument) are applied to first ntuple
  - New functions with ntupleId argument added for both creating ntuple and colums and also for the ntuple accessors, FillNtupleTColumn() and AddNtupleRow()
- No migration of user code with 1 ntuple is required

Handling More Ntuples (2)

- Depending on the selected output format more files can be generated:
  - Root, Hbook: a single file
  - Xml, Csv: each ntuple is written in a separate file
    - The ntuple file name is generated automatically from the base file name set via SetFilename() function or /analysis/setFileName command:
      - fileName_ntupleName.xml
      - fileName_ntupleName.csv
    - The histograms (if present) are written in a separate file (even if only one ntuple is defined):
      - fileName.xml
- See examples/extended/analysis/AnaEx01 where the ntuple was split in two

User Defined and Logarithmic Binning Scheme

- Request for ReverseMC example
- User defined binning scheme (available in g4tools) can be defined using new methods CreateH1[H2]() or SetH1[H2]() with
  - const std::vector<G4double>& edges instead of
  - G4int nbins, G4double xmin, G4double xmax arguments
- Logarithmic binning scheme can be selected using new argument in CreateH1[H2]() or SetH1[H2]():
  - const G4String& binSchemeName
    - When "log" value is selected (default is "linear"), the edges for logarithmic binning are automatically computed
- Selection is also available via UI set and create commands
  - /analysis/h1/set id nbins xmin xmax [unit] [fcn] [binScheme]
  - Analogous command also for h2

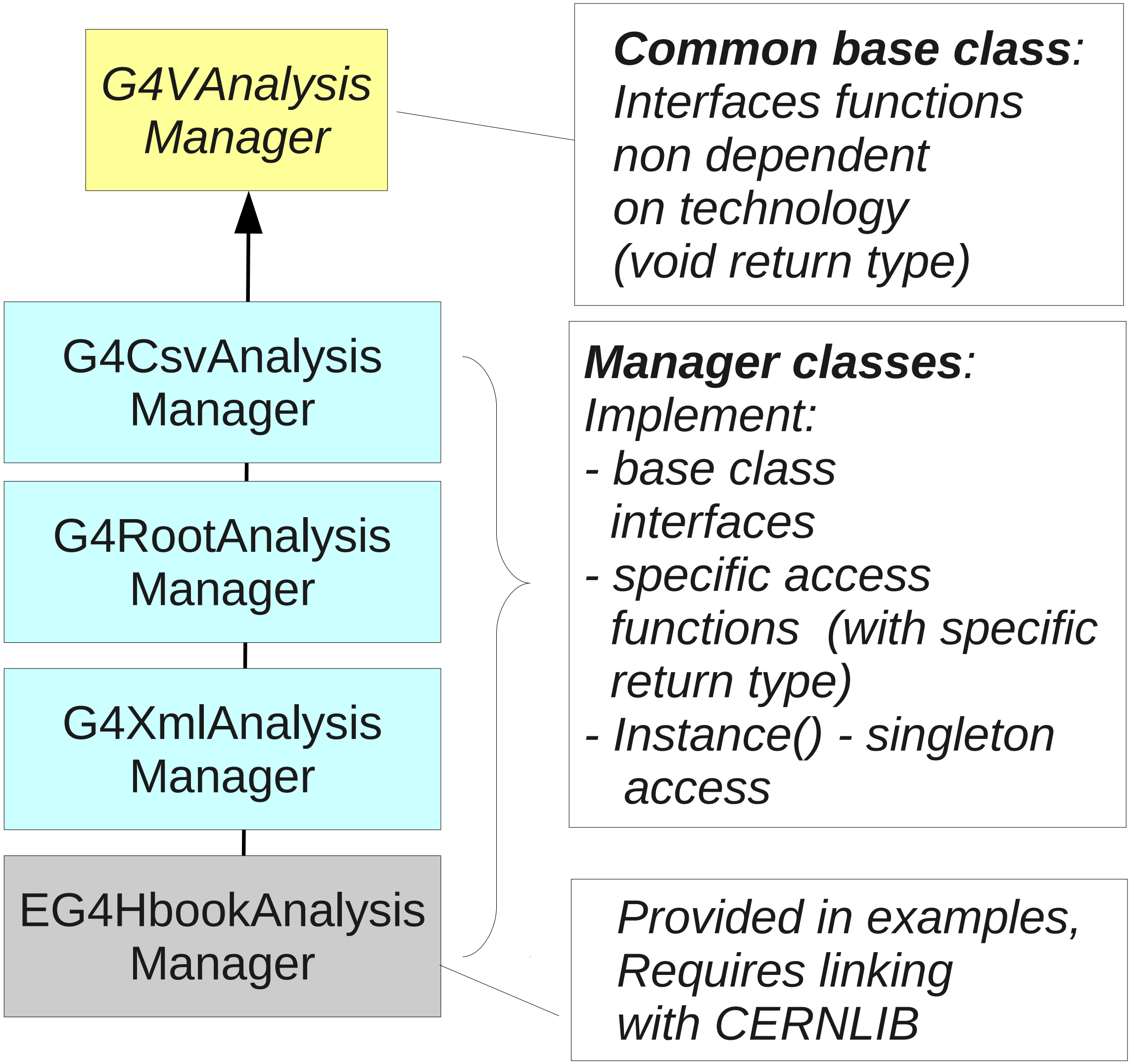
Migration to Multi-threading

- The analysis manager instances are created on master and thread workers
- New function to instatiate the analysis manager:
  - G4AnalysisManager::Create(G4bool isOnMaster)
  - With recently added functions in G4Threading namespace isOnMaster may be detected automatically and the code can be probably simplified for 10.00
- No other changes required in the user client code
- Locking via G4AutoLock & mutexes
  - For H1 and H2 merge (necessary)
  - For instances counter: this counter was added as a work-around for non existing functions providing information about the executed mode; to be revised

Migration to Multi-threading (2)

- Depending on the output format, the instantiated objects and selected Geant4 mode (sequential or multi-threading), several files can be created with automatically generated names
  - Histograms produced on thread workers are merged on Write() call and the result is written in the master file
  - Ntuples produced on thread workers are written on separate files. No merging is performed.
    - The ntuples files names are automatically appended with “_t” followed by thread Id (0, 1, 2, ...):
      - fileName[_ntupleName]_tid.xml
- Files clean-up:
  - Empty files (eg. If no ntuple is defined) are removed on CloseFile() call

Analysis Managers
First Design



***Common base class**:*
Interfaces functions non dependent on technology (void return type)

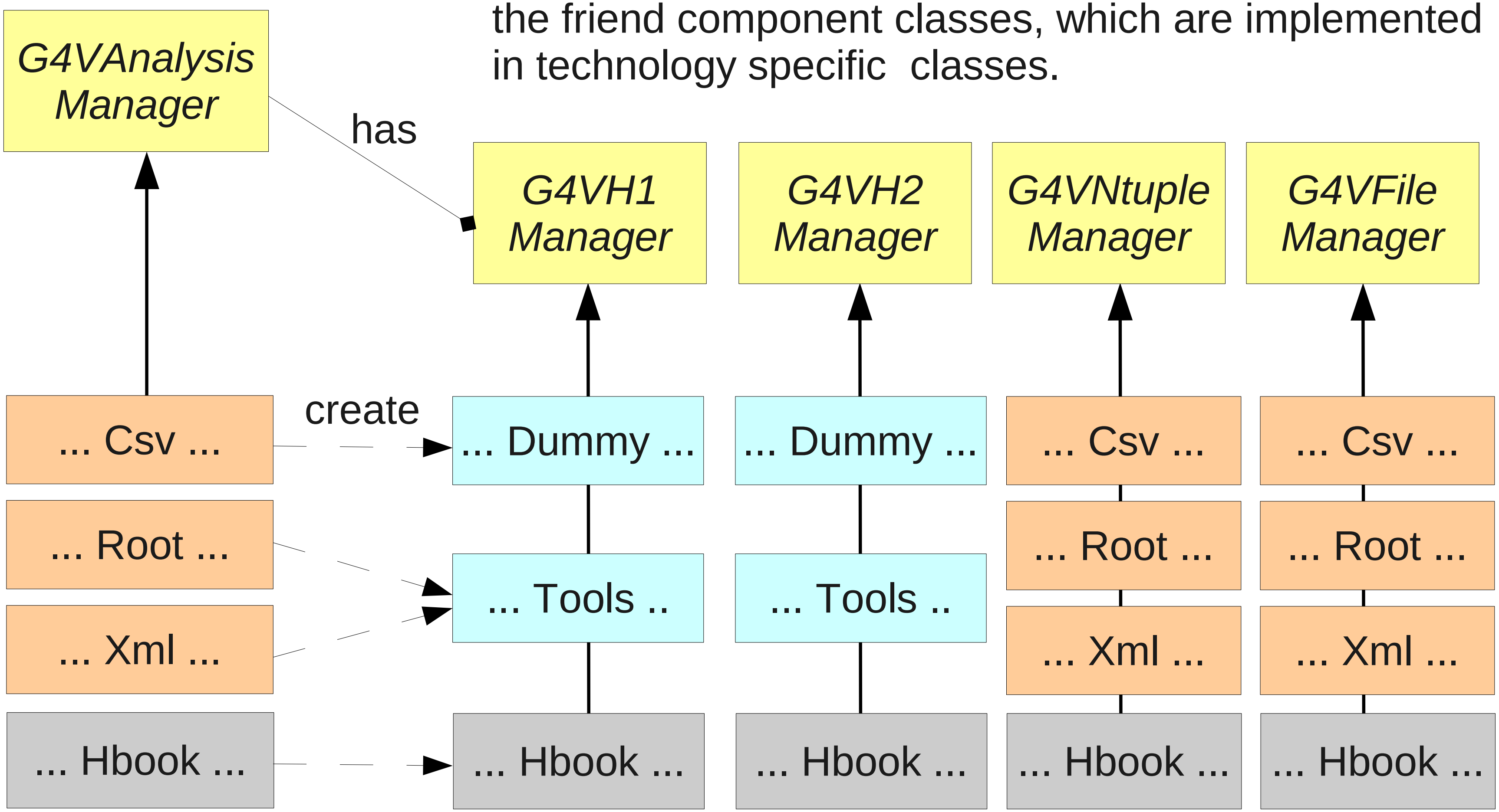
***Manager classes**:*
Implement:
- base class interfaces
- specific access functions (with specific return type)
- Instance() - singleton access

Provided in examples, Requires linking with CERNLIB

- Separating the manager classes per object type: h1d, h2d, ntuple - in 2013 plan
- To reduce the classes size
- To avoid code duplication
  - Some manager classes handles the same tools object type; eg. h1d, h2d in XML and Root managers

New Design

NVI (Non Virtual Interface) pattern: Non virtual public methods call protected, pure virtual members in the friend component classes, which are implemented in technology specific classes.



Analysis Category

- In source/analysis
- Restructured in sub-directories:
  - g4tools: ➢ include/tools
  - management: ➢ interface classes and utility classes
  - cvs, root. xml: ➢ classes specific to the output format
  - hntools: ➢ classes for h1, h2 management (shared by Root and Xml managers)
  - test: ➢ standalone tools tests (not using Geant4 code)
- Optional Hbook manager in
  - examples/extended/common/analysis

g4tools

- Behind the analysis managers there are the tools classes which do the real work
- Developed and maintained by Guy Barrand, LAL
- From version 1.4.0 in 9.6 to 1.7.0 in ref-09
- See History_tools file (in source/analysis) for detailed description of all changes

Plans

- Support 1D, 2D profiles
- Continue addressing new requests from users

Conclusions

- The analysis category is used in examples since 2 years
- The users questions in hyper-news & 2 bug reports show also the interest of Geant4 users
- Migration of remaining extended & advances examples planned for 10.00 release
  - New requirements from Geant4 developers were addressed
  - One example based on AIDA is kept
- Many thanks to Guy Barrand for prompt responding and fixing problems in tools