AIDA analysis tools and Geant4
A user roadmap

G.Barrand / LAL / IN2P3 / CNRS

FAQ : what is AIDA ?

- Abstract Interfaces for Data Analysis.
- A lot of things, but for a user it presents itself as a user API for doing statistical analysis, especially manipulating common things like histograms and tuples.
- API defined by HEP people of CERN, LAL, SLAC covering now three OO languages (C++, java, Python)

- (A lot of “HEP sociology” too…)

FAQ : I download AIDA and nothing work ?

- Then this “API”, due to the magic of OO languages can be implemented in the form of “pure abstract interfaces”(*) for these languages (C++, java, Python for the moment).
- On the web site : http://aida.freehep.org there is only the description of the “API”, then the “interfaces” .
- But there is no implementation here !
- Implementation must be taken from “AIDA providers”.

* In C++, a class with only purely virtual methods (not the same than a « base » class)

FAQ : providers ?

- C++ : OpenScientist (*)
  - http://OpenScientist.lal.in2p3.fr
  - AIDA implementation plus OpenPAW and Lab interactive tools.
- C++ : around CERN (a lot of sociology. No comment (**)) :
  - Anaphe : http://anaphe.web.cern.ch
  - PI : http://lcgapp.cern.ch/project/pi
- Java : http://java.freehep.org
  - JAIDA : then in java.
  - AIDA/JNI : a C++ front end to JAIDA.
  - Jas interactive tool
- Python : PAIDA : http://paida.sourceforge.net

* this presentation is totally, definitely, absolutely OpenScientist biased

** only at coffee breaks

FAQ : my histograms in files ?

- First need : write a “batch” program able to create histograms, fill them and put them in files workable later by various “interactive” tools.
- Most of providers can do that, but look closely at their documentation about the supported “exported” file formats. In particular if targeting a particular interactive tool.
- Promoted : .aida : an xml file format for the “data” classes supported in the AIDA API : IHistogram, ITUple, ICloud, IDataPointSet.
- OpenScientist-15 writes .aida, .ascii, .hbook, .root

```
#include <AIDA/AIDA.h>

#include <stdlib.h>

#include "Random.h" //Not AIDA

int main() {
  // The main and only concrete entry point !
  AIDA::IAnalysisFactory* aida = AIDA_createAnalysisFactory();
  if(!aida) return EXIT_FAILURE; //Bad luck. Contact your AIDA provider...

  // Someone get everything through factories...
  AIDA::ITreeFactory* trf = aida->createTreeFactory();
  if(!trf) return EXIT_FAILURE;

  // Create a tree-like container associated to a file...
  AIDA::ITree* tree = trf->create("myFile.aida","xml",false,true);
  //Depending of implementations : "ascii", "hbook", "root".
  delete trf;
  if(!tree) return EXIT_FAILURE; //File format not supported ?

  AIDA::IHistogramFactory* hf = aida->createHistogramFactory(*tree);
  if(!hf) return EXIT_FAILURE;

  // Ah ! my histo at last...
  AIDA::IHistogram1D* h = hf->createHistogram1D("test 1d",50,-3,3);
  delete hf;
  if(!h) return EXIT_FAILURE;

  // Well, do something with the histo...
  Random r;
  for (int i=0; i<10000; i++) h->fill(r.nextGaussian());

  tree->commit();  // Write to file...

  delete aida;  // Bye, bye the AIDA implementation...

  return EXIT_SUCCESS; //Yeee...
}
```

FAQ : tuple ?

```
AIDA::ITupleFactory* tf = aida->createTupleFactory(*tree);
if(!tf) return EXIT_FAILURE;

// My AIDA tuple...
AIDA::ITuple* tuple =
  tf->create("myTuple","My tuple","double v1, double v2");
if(!tuple) return EXIT_FAILURE; //Allo ?
delete tf;

// Fill it...
for(int i=0;i<10;i++) {
  tuple->fill(0,(double)i);
  tuple->fill(1,(double)2*i);
  tuple->addRow();
}
```

FAQ : installation ?

- See the provider doc…
- A general comment : a data analysis system (whatever it is) is something de facto complicated since it has to cover ALL computing domains (storage, graphic, GUI, scripting,…).
  - DO NOT EXPECT A STRAIGTFORWARD INSTALLTION (*)
  - KEEP COOL
- OpenScientist-15 : a lot of effort done on installation. Now a "clickable installation" for Windows and Mac. rpm for SLC3.

* Compared to, installing Geant4 kernel is a piece of cake…

FAQ : setup, compile, link ?

- "setup" the implementation with the aida-setup scripts :
  - csh> source <aida_imp_path>/aida-setup.csh
  - sh> . <aida_imp_path>/aida-setup.sh
  - DOS> call <aida_imp_path>\aida-setup.bat
- On UNIX, compile and link with the aida-config program :
  - UNIX> g++ `aida-config --cflags` Store.cpp `aida-config --libs`
- On Win-d'Oz by using the AIDA_CONFIG_[CFLAGS,LIBS] environment variables :
  - cl.exe %AIDA_CONFIG_CFLAGS% Store.cpp %AIDA_CONFIG_LIBS%
- An advice : before dealing with AIDA in Geant4, check that a simple AIDA example, like the previous one, works for you.

FAQ : and for G4 examples ?

The G4 make system wants :

- G4ANALYSIS_USE
- G4ANALYSIS_AIDA_CONFIG_CFLAGS
- G4ANALYSIS_AIDA_CONFIG_LIBS

```
csh> setenv G4ANALYSIS_AIDA_CONFIG_CFLAGS `aida-config --cflags`
csh> setenv G4ANALYSIS_AIDA_CONFIG_LIBS `aida-config --libs`
csh> setenv G4ANALYSIS_USE 1
 sh> export G4ANALYSIS_AIDA_CONFIG_CFLAGS=`aida-config --cflags`
 sh> export G4ANALYSIS_AIDA_CONFIG_LIBS=`aida-config --libs`
 sh> export G4ANALYSIS_USE=1
DOS> set G4ANALYSIS_AIDA_CONFIG_CFLAGS=%AIDA_CONFIG_CFLAGS%
DOS> set G4ANALYSIS_AIDA_CONFIG_LIBS=%AIDA_CONFIG_LIBS%
DOS> set G4ANALYSIS_USE=1
```

examples/extended/analysis/AnaEx01

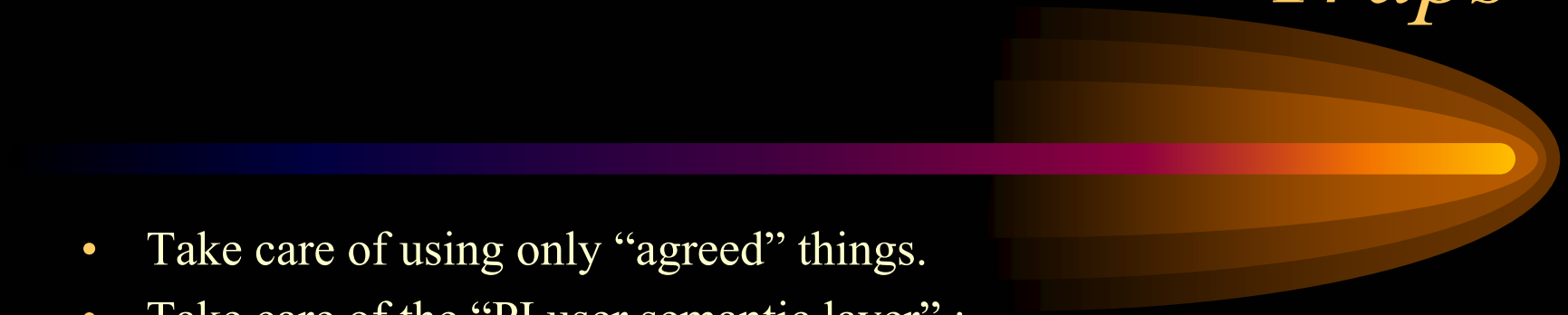
```
#include "G4RunManager.hh"
...
#ifdef G4ANALYSIS_USE
#include <AIDA/IAnalysisFactory.h>
#endif
...
#include "AnaEx01AnalysisManager.hh"
#include "AnaEx01DetectorConstruction.hh"
...
int main(int,char**) {
  G4RunManager * runManager = new G4RunManager;
  runManager->SetUserInitialization(new AnaEx01DetectorConstruction);
  ...
  AnaEx01AnalysisManager* analysisManager = 0;
#ifdef G4ANALYSIS_USE
  AIDA::IAnalysisFactory* aida = AIDA_createAnalysisFactory();
  analysisManager = new AnaEx01AnalysisManager(aida);
#endif
  runManager->SetUserAction(new AnaEx01RunAction(analysisManager));
  runManager->SetUserAction(new AnaEx01EventAction(analysisManager));
  runManager->SetUserAction(new AnaEx01SteppingAction(analysisManager));
  ...
  runManager->Initialize();  //Initialize G4 kernel
  ...
  G4UImanager* UI = G4UImanager::GetUIpointer();
  UI->ApplyCommand("/control/execute run.mac");
  ...
  delete runManager;
#ifdef G4ANALYSIS_USE
  delete analysisManager;
  delete aida;
#endif
  return 0;
}
```

examples/extended/analysis/AnaEx01

- It has an “AnalysisManager” to gather the bookings.
- No singleton (contrary to A01 or other examples) (*).
- It produces a AnaEx01.aida xml file (**).
- analysis/Lab/AnaEx01.py to show simple manipulation from Python.

* I hate singletons and static objects (put in shared libs)
** The default in next Geant4 release

Traps



- Take care of using only “agreed” things.
- Take care of the “PI user semantic layer” :
  - To book an histo by doing : PI:Histogram1D h(“h”,100,0,1);
  - This C++ API coming from CERN is not yet supported by the AIDA group and works only with one CERN implementation.
  - It breaks the C++ / java symmetry (and the SLAC team is NOT interested in having the same in java).

- Avoid “TClass::Draw” kind of software ; then the ones in which people confuse, for example, visualization and introspection and for which the sociology is such that there is no way to change the engineering of things.

Work plan

- Today AIDA-3.2.1
- AIDA-3.3 released :
  - OSC not ready yet.
  - Templated ITupleColumn : it will boost the tuple filling / reading and open AIDA to storing user data.
  - IGenericFactory : to open the system to user defined material.
  - IPlottable : to open the plotting to user data visualization.
  - IPlotter styles : more material but a lot have still to be done…
- AIDA-4.0 :
  - Exceptions
  - ISession
  - A “user semantic layer” coming from the AIDA group ?
  - AIDA::Dev interfaces : interfaces for the software engineer.
  - Etc…
- Support the HDF5 binary format in a compatible way between implementations.

Work plan around OpenScientist

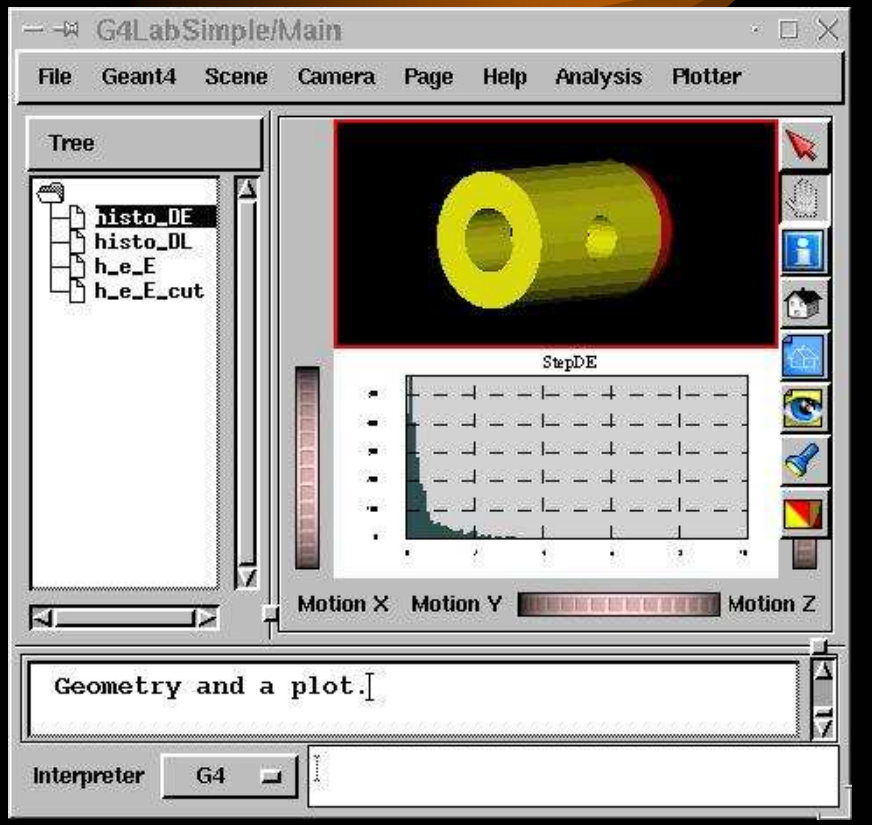
- AIDA-3.3
- HDF5
- Plotting styles
- Improve everything
- There is always something to do around a data analysis tool…

Work plan around Geant4 ?

- Follow AIDA releases.
- /aida G4 commands ?
- /aida/plot <the physics tables>
- Integrate AIDA plotting and G4 visualization in a consistent GUI.
- Strongly move toward python and Qt ?
- ???

Since it is OpenScientist biased…
G4Lab / UIOnX

- With the OpenScientist G4Lab package, someone can already integrate an Inventor visualization for Geant4 along with AIDA plotting in a consistent GUI.
- G4Lab::UIOnX : a G4 UI session for OnX.
- G4Simple : a simple application.
- G4SPL : a user application.
- onx -newapp -template=G4LabTemplateOne
- G4Examples : a package with some of G4 examples (N02-N07, AnaEx01, A01, gammaray_telescope) using G4Lab::UIOnX.
- See OpenScientist web pages.



Conclusions

- A lot exist now around AIDA and things improve. See the web sites of providers.
- It is probably the less intrusive way to do statistical analysis, and this by covering a lot of technologies in a consistent way.
- Somewhere, AIDA is unique.
- A bit of sociology : a pity that AIDA is systematically killed by some around CERN. Interfaces are a federating place at the level of the code, they should be promoted by a lab created to federate engineering sociology to do physics.
- And Geant4 should do the same… (some G4VXxx should be Geant4::IXxx)