HippoDraw Application and Library

Paul F. Kunz

Stanford Linear Accelerator Center

Brief overview of HippoDraw

Use from Python

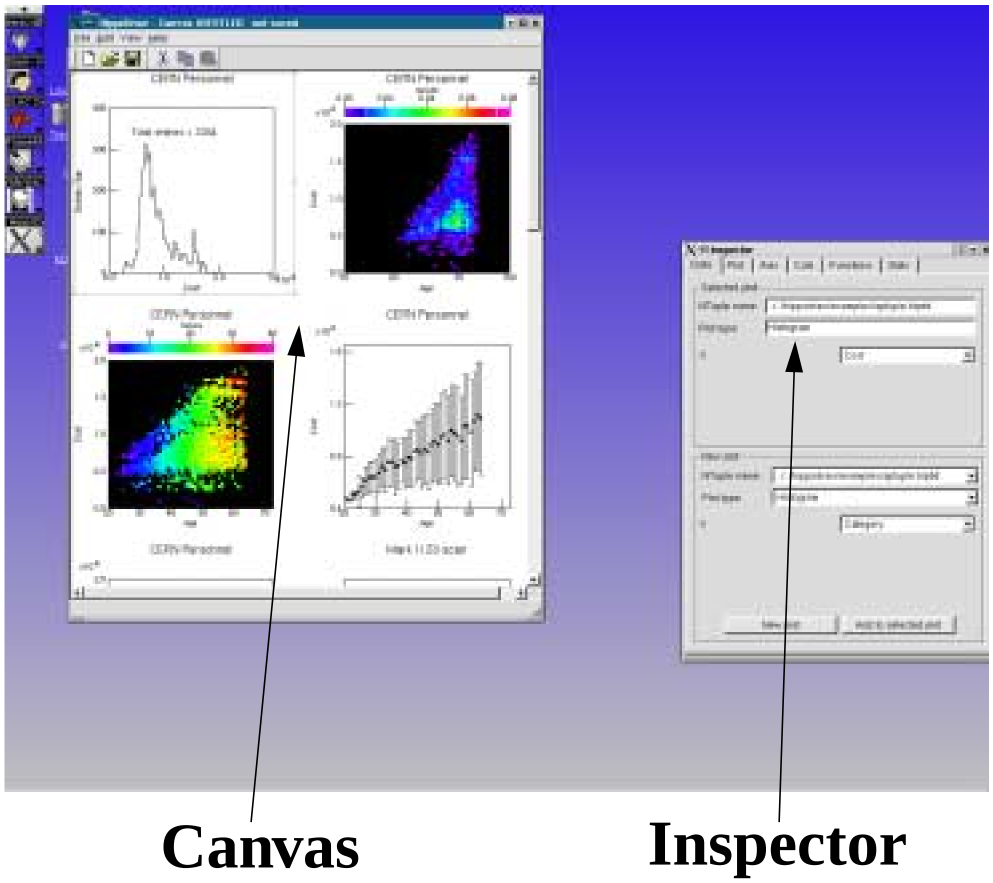
Use of library for custom applications

Two Versions
- Pure C++ version uses Qt
- Java/C++ version uses swing

Demonstration on 850 MhZ P3, 256 MB memory

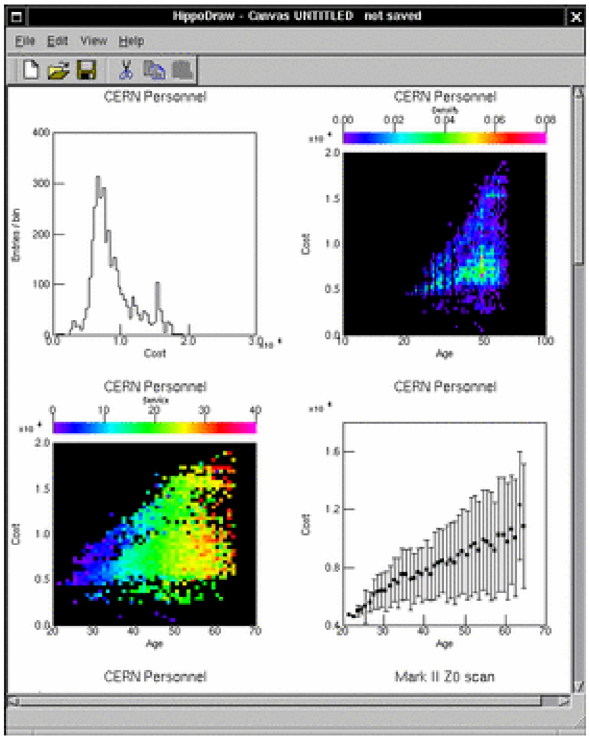
What is HippoDraw

An analysis package...



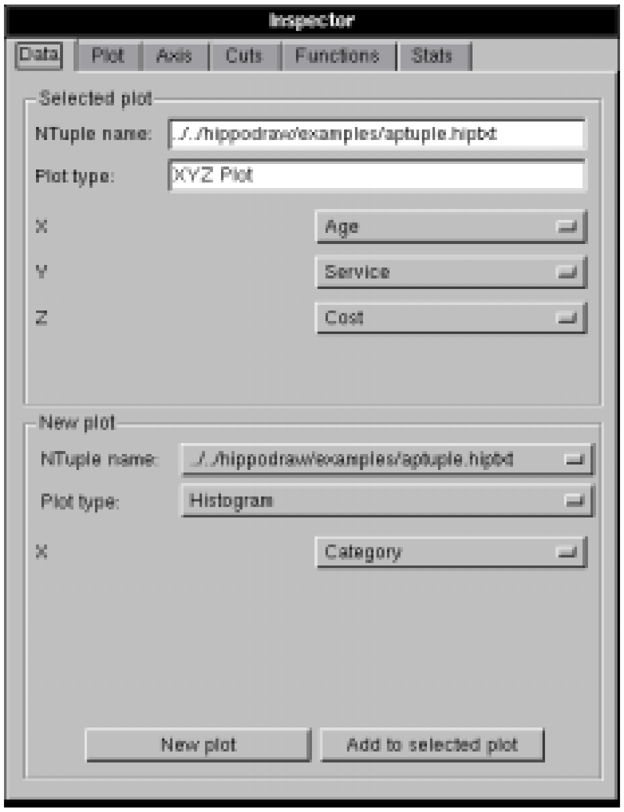
- Canvas contains the displays
- Inspector allows you to view properties and change them.
- The only windows except for modal dialogs

Document paradigm



- Canvas can be saved as multi-page document in XML format
- Documents can be opened at a later time
- Multiple opened documents are allowed
- One document serves as template for multiple data sets
- Eliminates one need for scripts

Data Inspector



Selected plot

NTuple name: ../../hippodraw/examples/aptuple.hiptxt

Plot type: XYZ Plot

X: Age

Y: Service

Z: Cost

New plot

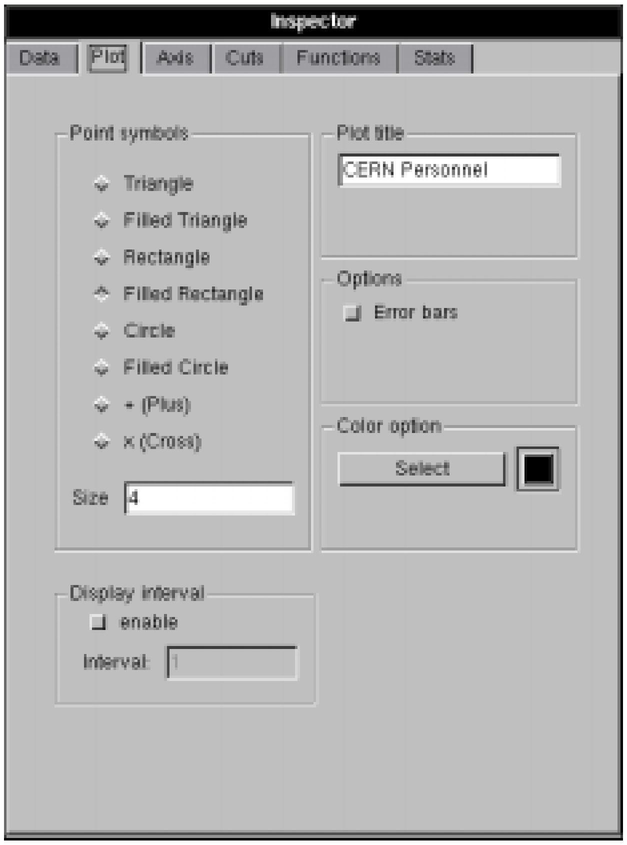
NTuple name: ../../hippodraw/examples/aptuple.hiptxt

Plot type: Histogram

X: Category

- controls creation of displays
- controls data binding
- GUI enquires to C++ DataRepFactory allows for extensibility

Plot Inspector



Point symbols
- Triangle
- Filled Triangle
- Rectangle
- [✓ Filled Rectangle]
- Circle
- Filled Circle
- + (Plus)
- x (Cross)

Size: 4

Plot title: CERN Personnel

Options
- ☐ Error bars

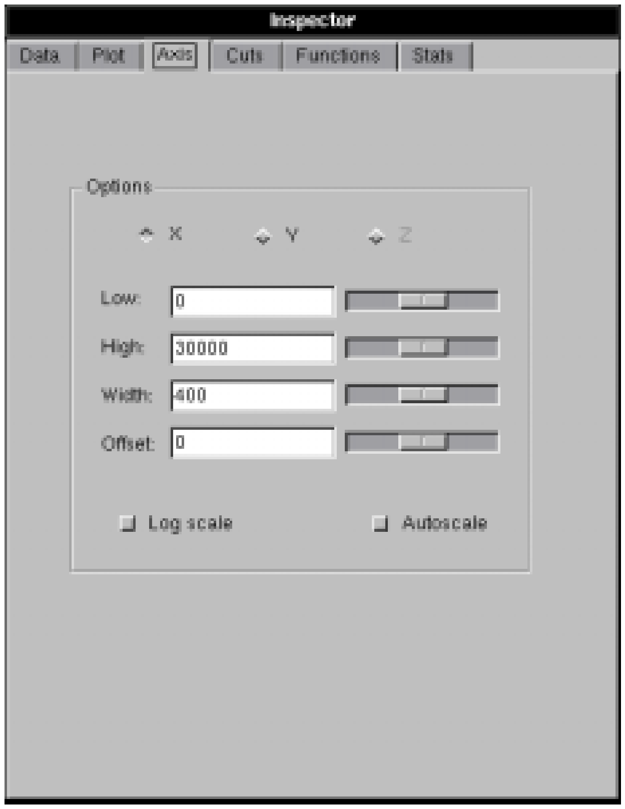
Color option: Select ■

Display interval
- ☐ enable

Interval: 1

- controls a few display options
- Display interval is used with real time systems

Axis options Inspector

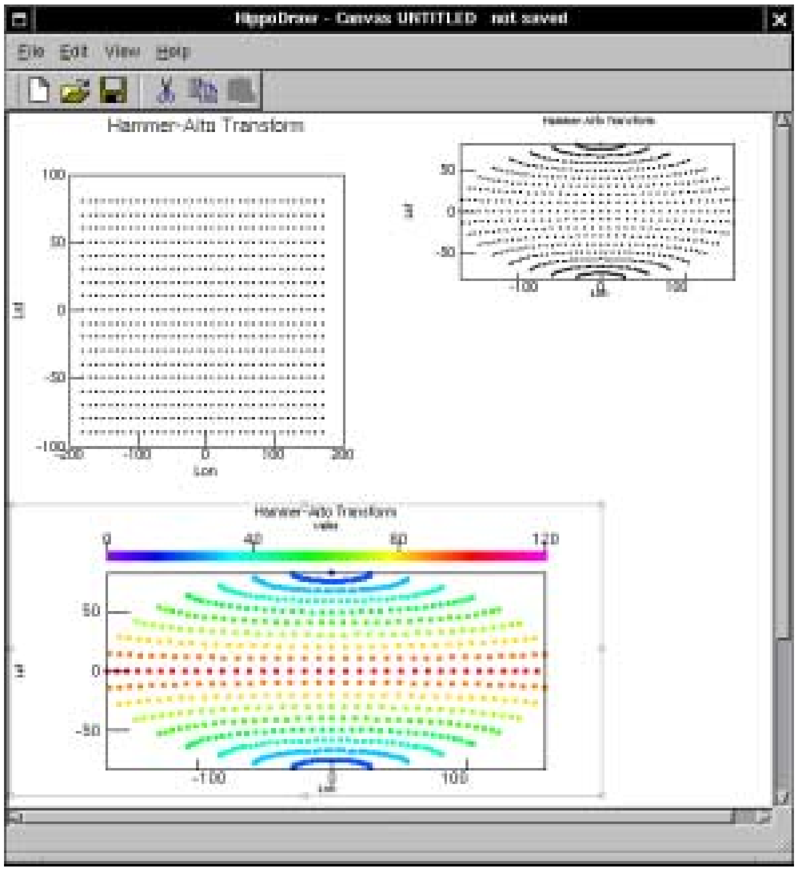


Low: 0
High: 30000
Width: 400
Offset: 0

Log scale
Autoscale

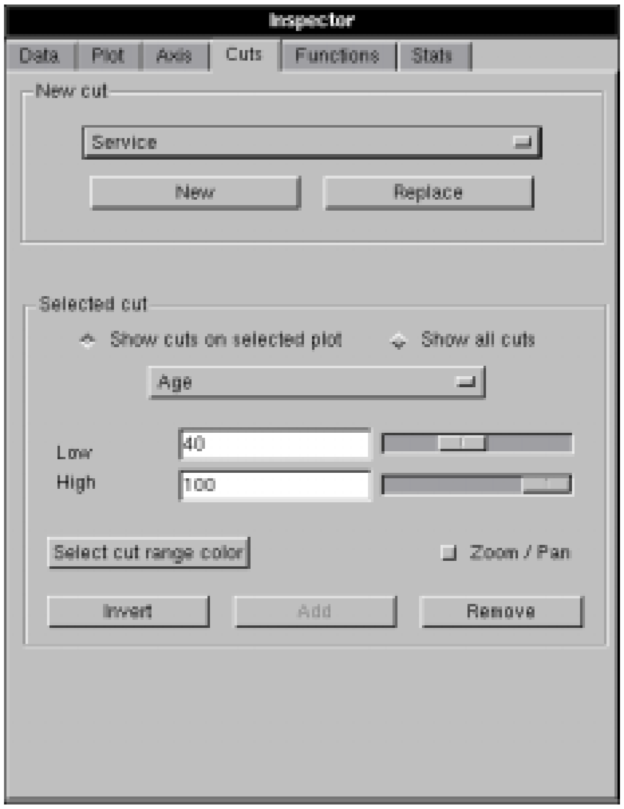
- controls axis range
- controls bin width and offset if binned
- note use of sliders
- log on X axis has logarithmic sized bins

Transforms



- Any display can be transformed
- lin-lin, lin-log, log-lin, log-log all supported
- The above is Hammer-Aitoff which must preserve aspect ration
- work in progress, not supported by GUI

Cut Inspector



Service

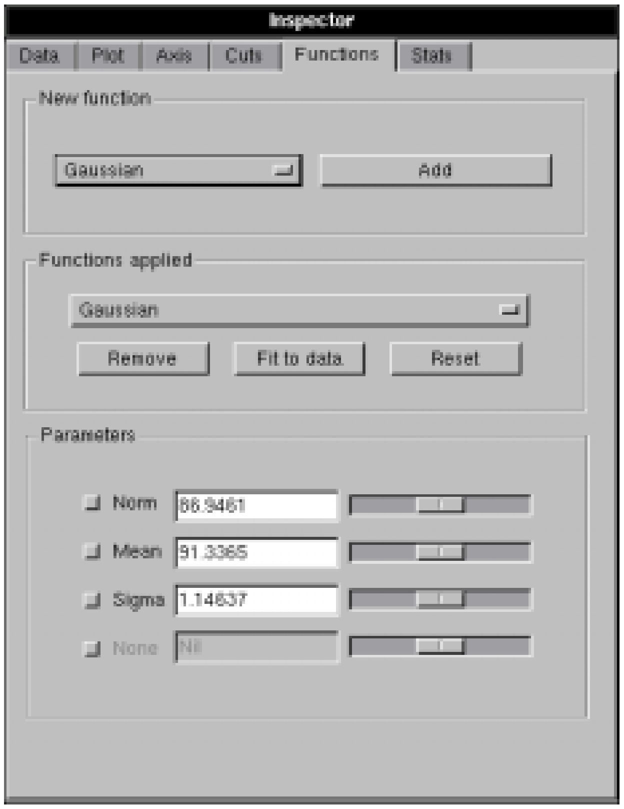
Age

Low 40

High 100

- controls creation and application of cuts
- cut range changed with sliders
- a cut can have multiple targets
- can use zoom/pan feature

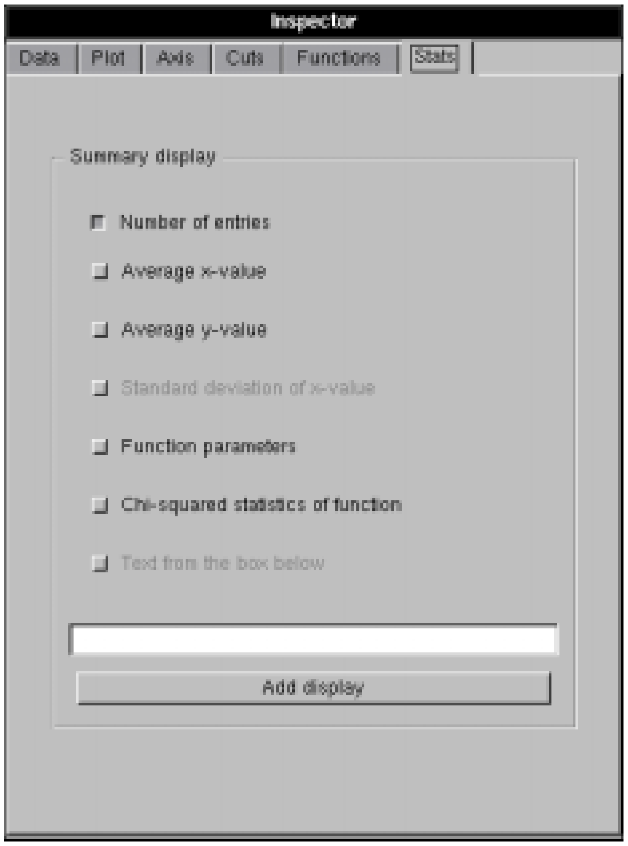
Function Inspector



Norm 86.9461
Mean 91.3365
Sigma 1.14637
None Nil

- controls creation and application of functions
- controls fitter
- GUI makes enquires to C++ FunctionFactory
- function parameter names from enquiry to C++ function objects

Summary Inspector



Summary display

- [✓ Number of entries]
- Average x-value
- Average y-value
- Standard deviation of x-value
- Function parameters
- Chi-squared statistics of function
- Text from the box below

- controls adding of textual representations
- the reps are “live”

Commands and scripts

HippoDraw can be used without commands or scripts

- ease of use is very good
- learning period is short
- to quote one CERN user: “*HippoDraw is so easy to use, even a 50 year old CERN physicist can use it*”

However, one needs a script to...

- do repetitive actions, *e.g.* 50 histograms on different channels
- massaging data
- reading special data formats
- getting and putting data from/to other packages

Solution: make HippoDraw a Python module

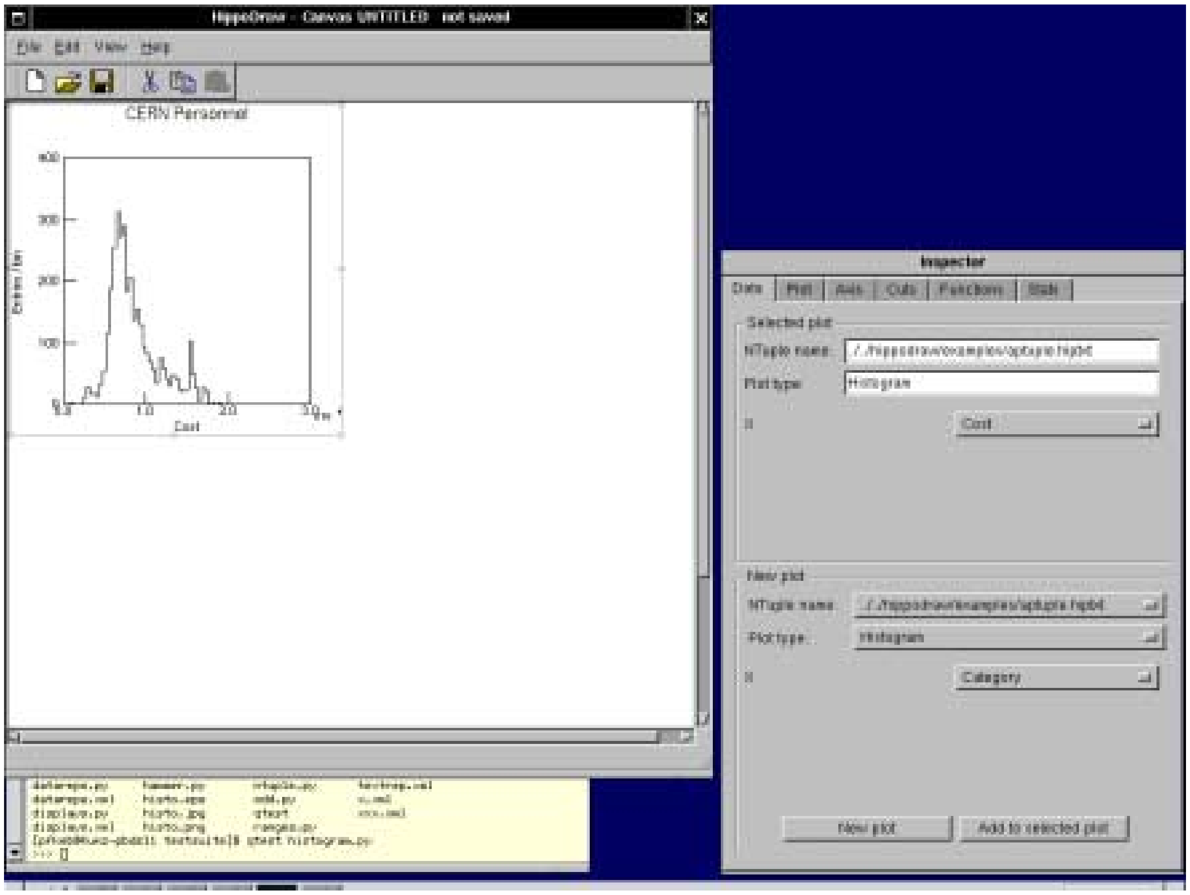
- HippoDraw becomes the non-intrusive slave to Python
- HippoDraw still does not have script language of its own

Simple Script

```
from hippo import *

app = HDApp()
canvas = app.canvas()

nt = NTuple ( ’examples/aptuple.hiptxt’ )

hist = Display ("Histogram", nt, ’Cost’ )
canvas.addDisplay ( hist )
hist.setRange ( ’x’, 0., 30000. )
```

- `hippo` is name of the Python module
- `HDApp`, `NTuple`, and `Display` are classes implemented in C++
- `app.canvas()` returns current canvas.
- `canvas.addDisplay()` adds display in next available free space

Result of script



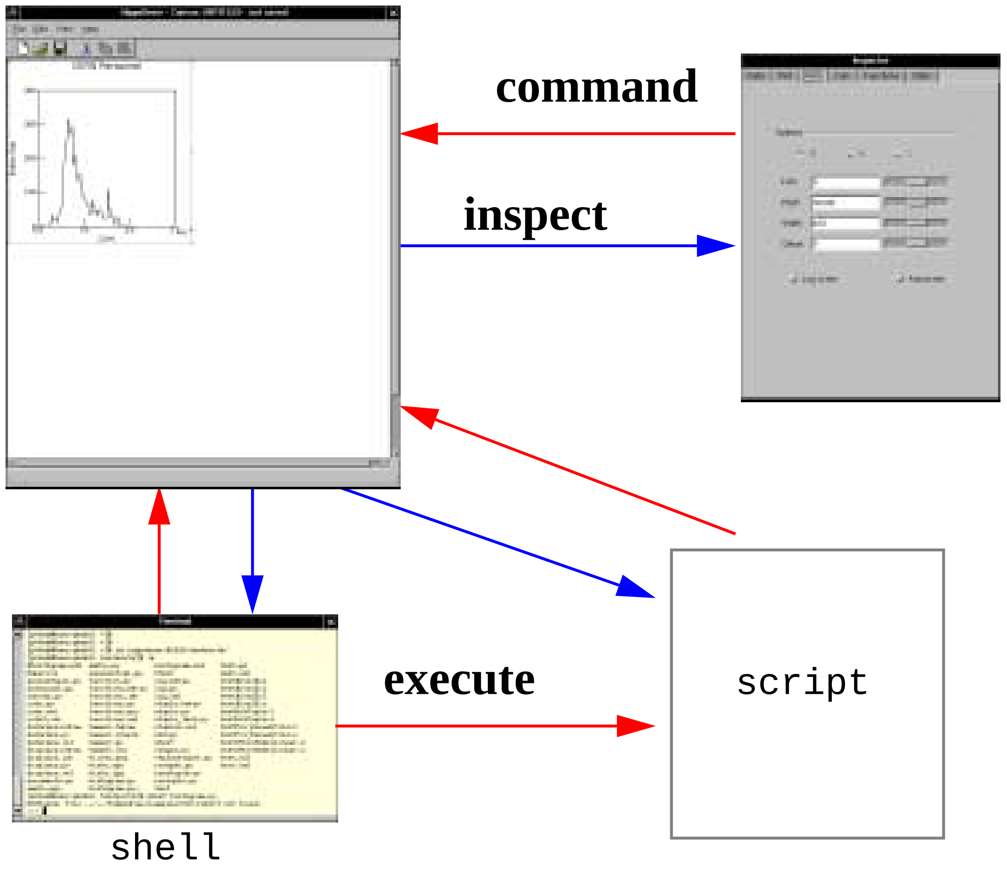
Selected plot
NTuple name: ../../hippodraw/examples/aptuple.hipbd
Plot type: Histogram
X: Cost

New plot
NTuple name: ../../hippodraw/examples/aptuple.hipbd
Plot type: Histogram
X: Category

```
datareps.py      hammer.py       ntuple.py       testrep.xml
datareps.xml     histo.eps       mdl.py          x.xml
displays.py      histo.jpg       qtest           xxx.xml
displays.xml     histo.png       ranges.py
[pfkeb@kunz-pbdb1 testsuite]$ qtest histogram.py
>>> []
```

- same as if one had used the GUI
- all GUI controls are active

Equal access



- Inspector can send commands and inspect canvas objects
- Python session or script can do the same
- they use the same member functions of the objects

Data access

In Python session or script

- create an empty ntuple (table of doubles)

```
nt = NTuple()
```

- add columns of equal length

```
nt.addColumn ( ’label’, array )
```

- add rows of equal size

```
nt = addRow ( array )
```

- can also replace row or column
- if ntuple used by displays changes, the displays update themselves immediately. Good for real-time applications

Example of massaging data

What you might do in Python

- Read data file with 100 channels of some measurement
- For each channel do a histogram
- Fit the histogram to, say, a Gaussian
- Extract the fit parameters
- add row to another ntuple with id, fit parameters, chi-squire, etc.
- create 3 XY Plots with id on x axis and a fit parameter on y axis
- apply cuts
- fit the XY plots to your model

You could do it interactively

- if you want to visually inspect each histogram and fit
- the XY plots will update with each addRow

You could write a script to do it “batch-like”

Data sources for Python

Python has many modules for reading data

Here are some...

- parse a file
- RPC library
- PyFITS (Astrophysics standard)
- RootPython (Pere Mato)
- Excel spreadsheet
- easy to roll your own (PAW?)

Other data sources...

- other Python modules, e.g. GaudiPython, PyGeant4
- algorithms implemented in Python
- HippoDraw ntuples, e.g. get data, massage, add new column

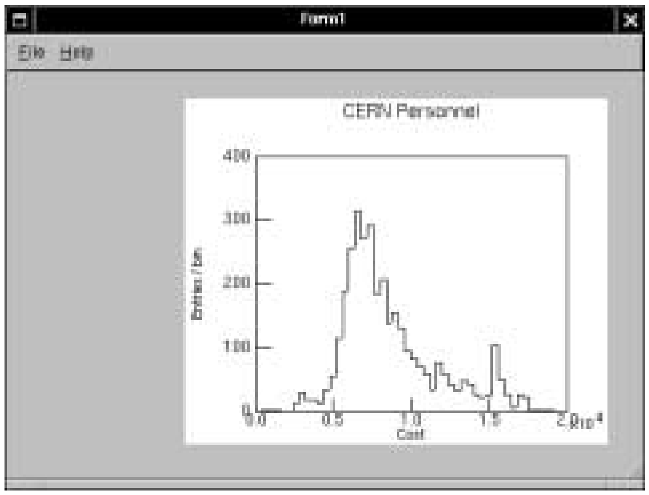
Python C++ interface

There are a number of them...

- boost.python and SIP
  - you write one line per constructor or member function.
  - capable of handling template classes like `vector<>` and `string`
  - HippoDraw has both boost.python and SIP
- boost.python is used for interactive mode as shown
  - friendly for script or interactive use
  - also used by GaudiPython, RootPython
- SIP is used for building applications with PyQt
  - programmer’s interface
- The interface is different, on purpose.

Custom Applications

HippoDraw library has Qt Widget as well as the Qt Canvas items already shown



- The above was made in Qt Designer with HippoDraw’s widget as a custom widget in Designer
- Can build custom applications with HiippoDraw widgets, canvas items, or both in C++ or Python
- Or can add HippoDraw canvas window and inspector to your non-GUI C++ or Python analysis job

The library

Design principle

- identify the steps going from raw data to a display
- each step abstracted and represented by a base class in a class hierarchy
- different ways to take a step are implemented in different derived classes
- A display is formed by selecting appropriate classes from each hierarchy

Decomposition

The steps

- create the n-tuple data: ***Ntuple*** 2 classes
- access n-tuple column(s): ***Projector*** 13 classes
- bin the data (optional): ***Bining*** 7 classes
  - uses binner: ***Binner*** 3 classes
- create projected values: ***Projected value*** 3 classes
- present projected values to point representation: ***Point representation*** 16 classes
- transform coordinate: ***Transform*** 7 classes
- draw axis, labels, *etc*.: ***Plotter*** 6 classes
- draw to graphics system: ***View*** 4 classes
  only dependence on Qt, Java, or OpenGL

- composite of Projector, Point rep, and optional binning: ***Data representation*** 13 classes

The library

The library consists of over 100 C++ classes

Java	Qt	OpenGL
application logic		
core		

- very modular
- easy to extend
- 35K lines of code, 2 MB binary (stripped), compiles in 3 minutes
- well documented (using Doxygen)
- the core is independent of the graphics system
- the core is independent of the application
- grounds up modern design

Such a library is open for

- experimentation on new data representations
- use in custom applications

Grubby details

Hippodraw compiles with...

- egcs 1.1.2 thru gcc 3.2.2 (including 2.96)
- VC++ 6.0 thru VC++ 7.0 (.NET)

Tested on...

- Solaris 5.8 (with gcc 3.1.1)
- Red Hat Linux 6.1, 7.x, 7.3.1, 8.0
- Mandrake 8.0 thru 9.0
- DESY SuSE 6.x
- Windows NT 4.0, 2000, and XP
- Mac OS X native and X11 (Python problems?)

On UNIX, Linux and Mac OS X-X11, uses free version of Qt to build

On Windows and Mac native, need Qt Enterprise license to build, but can distribute binaries royalty free

On Linux distributions with Qt 3.x, only external package is boost.python

Conclusions

HippoDraw as a stand-a-lone application offers the users great interactivity and document centric features.

HippoDraw as a module on the Python software bus effectively extends its usability to a much wider domain of applications

The HippoDraw library can be used in custom applications

The library is easily extendable for new kind of displays

Home page:

http://www.slac.stanford.edu/grp/ek/hippodraw/index.html