QtROOT
a Qt interface to ROOT

Denis Bertini
GO4 GSI-Darmstadt
13/06/2001

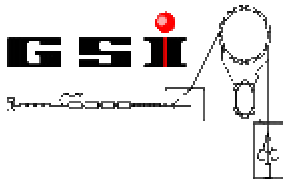
Outline

- Motivations
- QtROOT interface implementation
  - use of general ROOT GUI Factory classes
  - Integrating ROOT system events
- Ongoing work
- Conclusions

Motivations

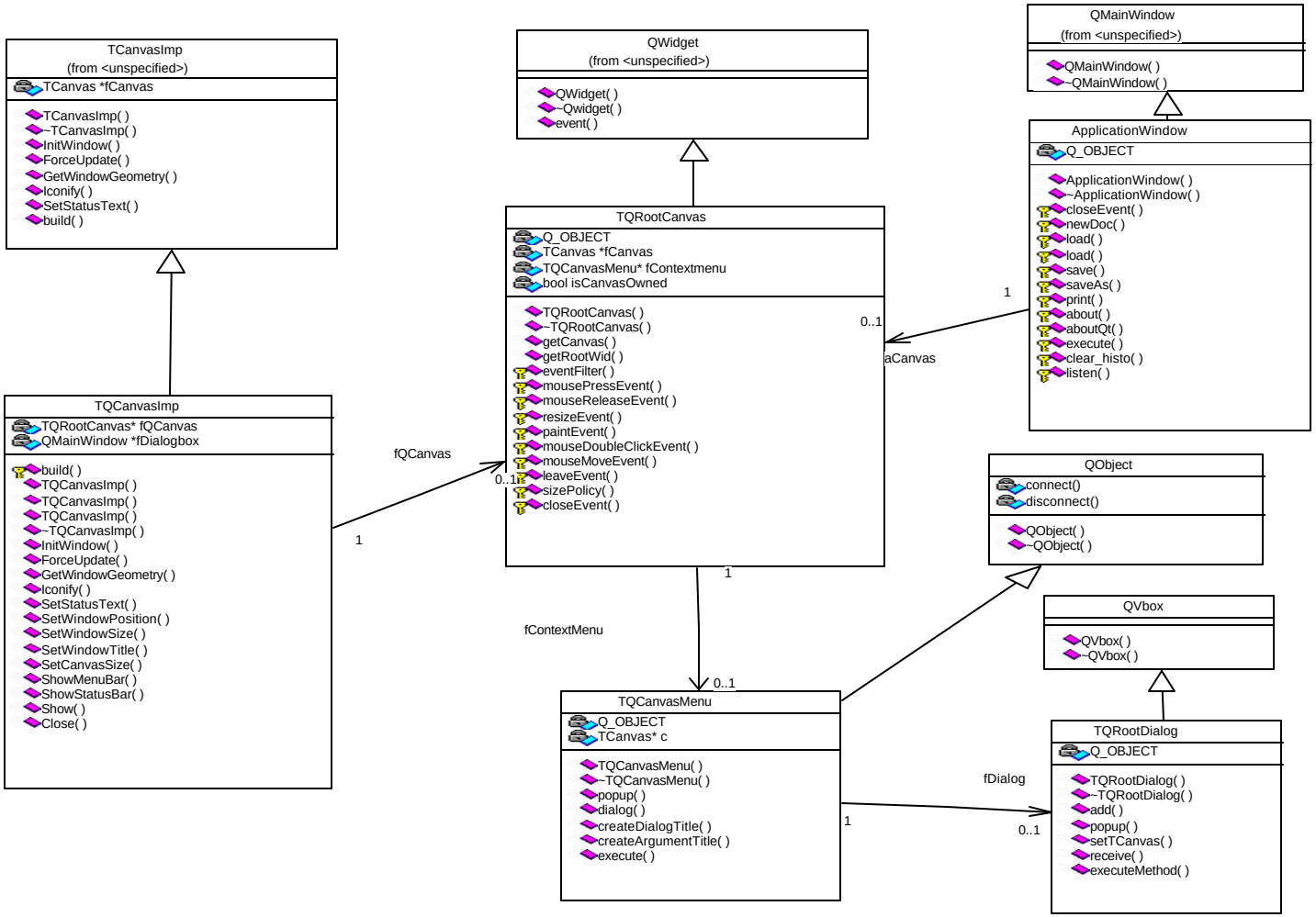
- To Give the ROOT user the opportunity to develop GUI application using together:
  - full embedded ROOT canvas functionality
  - full KDE/Qt cross-platform C++ class library
  - full ROOT functionality i.e ROOT system events available (TTimer, TSockets , TThread, …)
  - CINT and Qt?
- Visual design tool available : QtDesigner
  - compatible with QtROOT embedded canvas.

QtROOT components



- **Follows the general ROOT GUI ABC**
  - no changes in ROOT source necessary
- **Inheritance from ROOT base classes**
  - TQRootGuiFactory (TRootGuiFactory)
    - Qt Factory GUI components
  - TQCanvasImp (TCanvasImp)
    - creates a Qt independent main window (QMainWindow)
    - sets TQRootCanvas as a central widget
  - TQApplication (TApplication)
    - ROOT environment set with a Qt GUI factory
- **Inheritance from Qt base class**
  - QRootApplication (QApplication)
    - enable Qt evt-loop to drive Qt based GUI applications
    - call ROOT inner loop repeatedly to enable ROOT system events

QtROOT UML Diagram



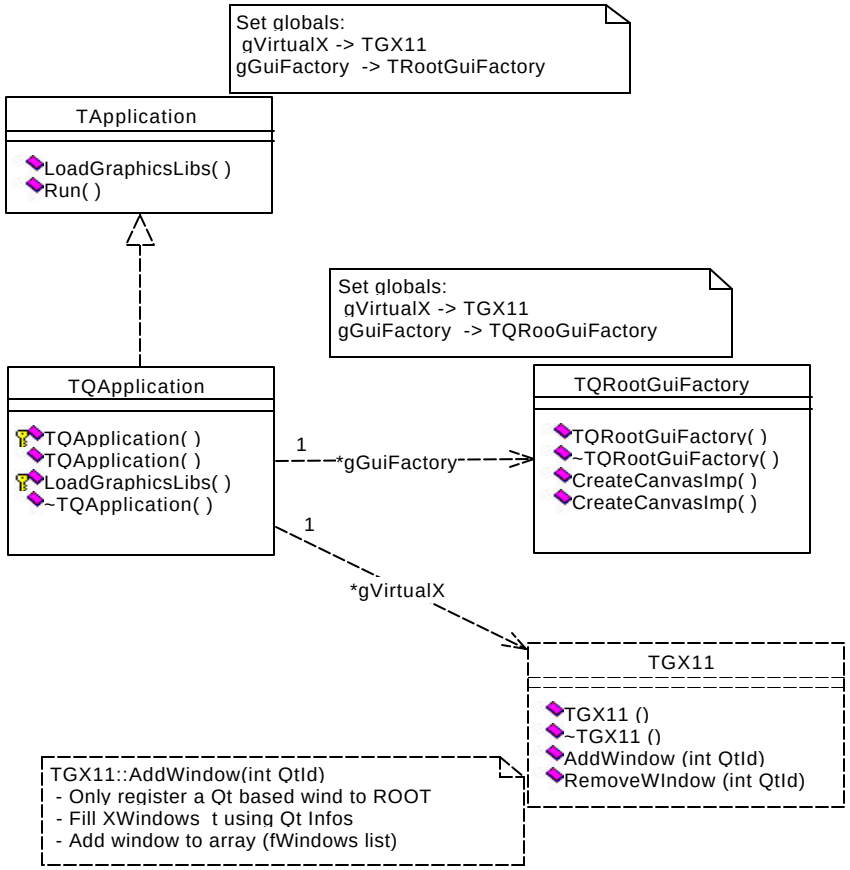
QtROOT GUI Factory

- TQApplication
  - LoadGraphLibs() sets global
    - gGuiFactory -> TQRootGuiFactory
- TRootGuiFactory
  - uses services from ABC TGuiFactory
  - enables usage of native Qt components + ROOT widgets
  - overriding:

- TGuiFactory::CreateCanvasImp(TCanvas *c, const char *title, UInt_t width, UInt_t height);
- TGuiFactory::CreateCanvasImp(TCanvas *c, const char *title, Int_t x, Int_t y, UInt_t width, UInt_t height)

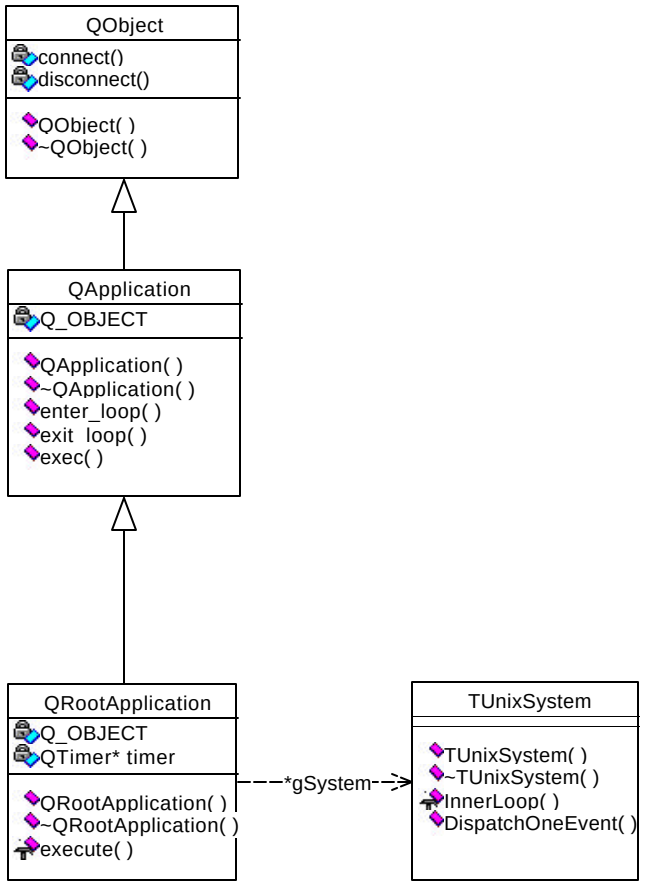
Services:

- Creates a specific Canvas Implementation TQCanvasImp



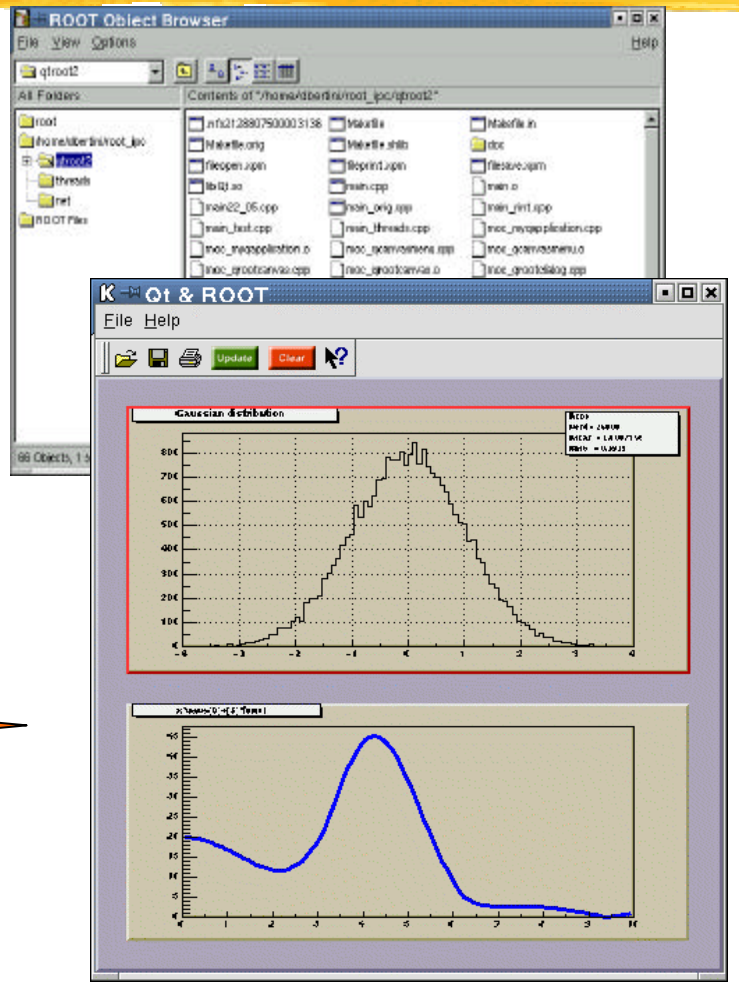
Integrating ROOT events

- QRootApplication
  - Qt event loop
    - uses QApplication::exec()
  - ROOT inner loop
    - connect itself to TUnixSystem::DispatchOneEvent() via a Qtimer class.
    - ROOT inner loop is the called repeatedly and return quickly
    - ROOT system event (TThreads, TTimers, Network classes) are enabled while Qt event loop is activated
    - ROOT Gui TGxx classes can also be used together with Qt Widget !



QtROOT Main program

```
Int main(int argc, char** argv){
  QRootApplication a (argc, argv );
  TROOT root( "QtRoot","test");
  TQApplication app("QtRoot",&argc,argv);
// Native ROOT GUI classes
  TBrowser b;
// Qt windows
  ApplicationWindow mw;
  mw.setCaption("QtRoot");
  mw.show();
  int res = a.exec();
  return res;
}
```



Ongoing work

- Further tests of QtROOT on Unix Platform
- GO4 Qt GUI front end in development
- HADES Qt based online monitoring
- Port to Qt3.0 (when release available)
- Investigate integration of OpenGL

Conclusions

- Full embbeded ROOT Canvas functionnality is available and tested
- ROOT system events integration has been tested within the multithreaded GO4 framework which uses TTimers, TThreads and ROOT network classes all together
- Running Qt widgets in parallel with ROOT widgets
  (e.g TBrowser) is possible
  (see Mohammad Al-Turany talk)