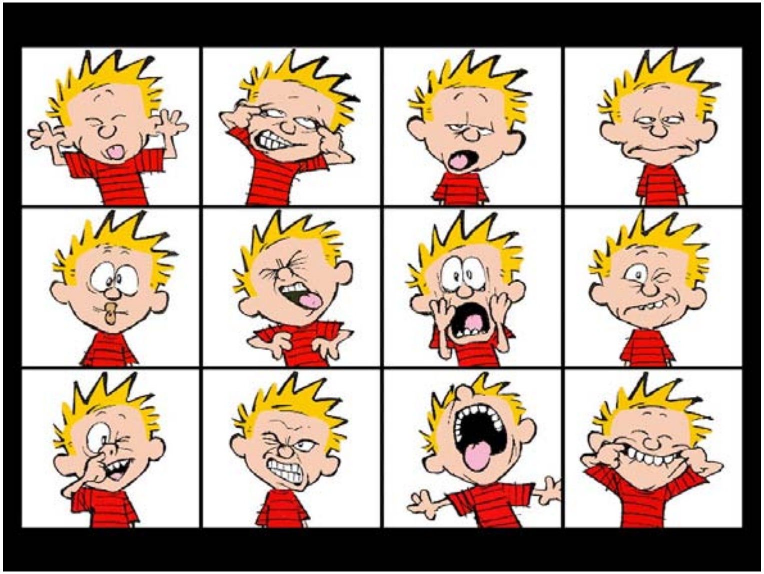
FRED, an Event Display Experiment

R.Giannitrapani

Roma, 15 September 2003



Dipartimento di Fisica
Università degli Studi di Udine
Sezione INFN Trieste-Udine

Contents

What's Fred?
Philosophy
HepRep production
What we got along the way?
Roadmap

And now for something completely different, *Monty Python*

This a short presentation on FRED (an acronymous that means **FRED is a Recursive Event Display**), a work in progress project to provide a flexible and modular Event Display suitable for GLAST and other similar experiments.

A big thank to **M.Frailis** (Udine), coauthor of FRED and **Joe Perl** (SLAC), author of the HepRep protocol on which FRED is based.

Slides made in ConTEXt, edited in emacs, rendered in PDF.

What's Fred?

Well, as most of you already know

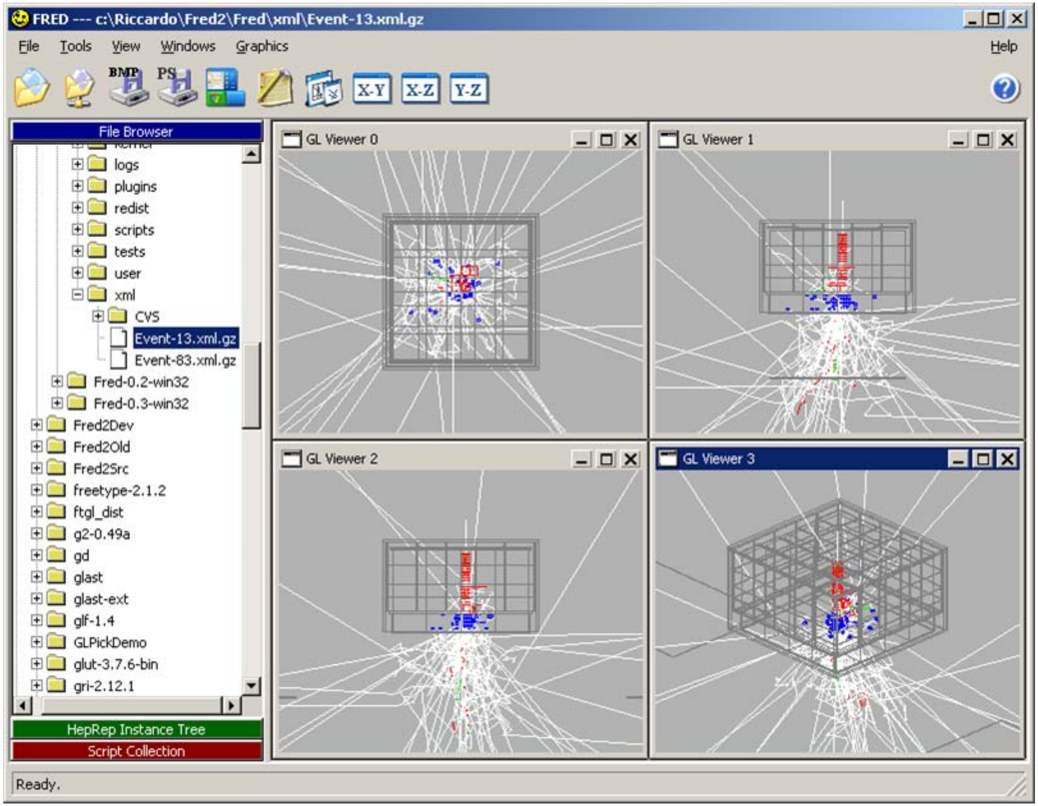
- It's an event display based on an abstract protocol, **HepRep**
  - Clear separation between graphics issues and physics ones
  - Well designed and flexible
  - Think at it as an HTML for event displays
  - Already in use in BaBar (WIRED) and other small projects
- It's an application, but with some twists of a **framework**
- Completely experiment independent, but can be customized for GLAST
- It's still a prototype, but already usable
  - Already presented and discussed in last GLAST software meetings and in conference (CHEP2003).
  - The last beta version (0.45) has been released during the Data Challenge Workshop at SLAC in July

Philosophy

Again, most of you already know:

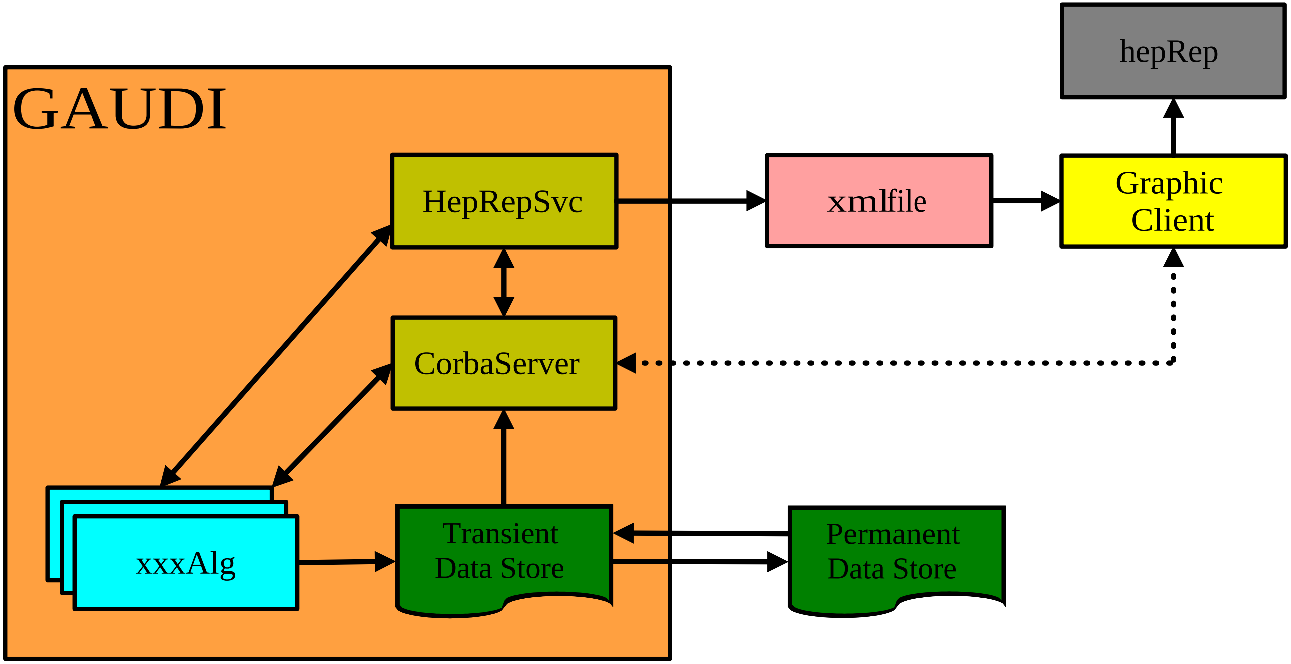
- **Avoid to commit (if possible) to a single product/application**
- **Be as much as possible flexible to accomodate future changes (there will be for sure)**
- **Let the user mess with the functionalities and customize it throught simple scripts**
- **Allow both 3D views and proper 2D views**
  - Use of industry standard **OpenGL**
  - Standard features and shapes in place, advanced features (layers, text) coming
- **Allow interactivity and browsing of the event**
- **Allow a batch (no gui) usage**
- **Allow a remote calling capabilities**

- Remote use and calls from other C++ applications
- We are using industry standard **CORBA**



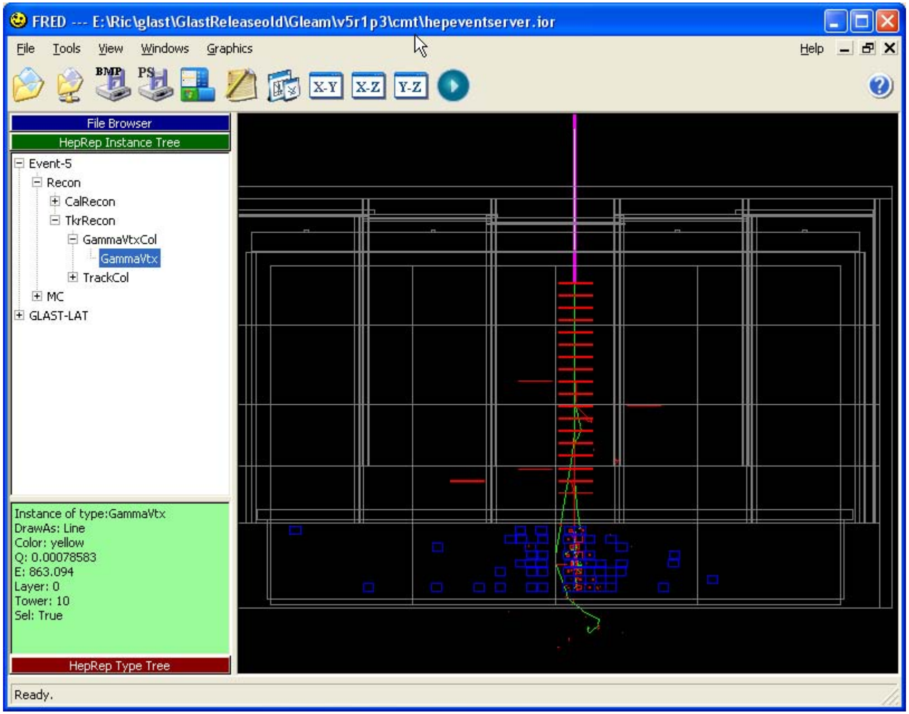
HepRep production

FRED is just a possible graphic client of a full HepRep framework



There are lot of technical items on the production of the HepRep representation of GLAST events that I don't discuss here. Summarizing ...

- We have a stable and well working GAUDI packages, **HepRepSvc** and **CorbaSvc**, to be used to produce both HepRep file and live HepRep CORBA server
- We have started the fillers for reconstructed data (to use FRED only with montecarlo data is boring); you can start to see right now the recon tracks and their relevant properties



Instance of type:GammaVtx
DrawAs: Line
Color: yellow
Q: 0.00078583
E: 863.094
Layer: 0
Tower: 10
Sel: True

What's Fred?
Philosophy
HepRep . . .
What we got . . .
Roadmap

What we got along the way?

Independently from the possible uses of FRED in future data analysis tasks, can we learn something from it for event display and, in general, for graphical applications and gui for GLAST?

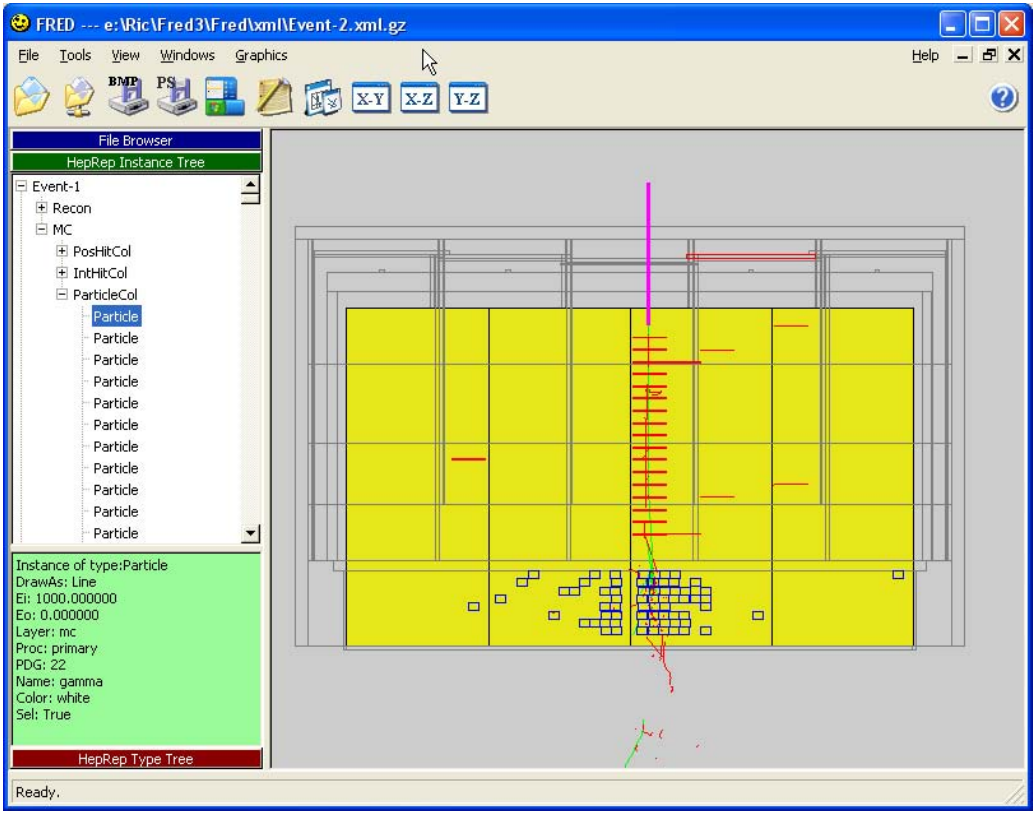
- ▷ **Use a scripting language ...**
  - Faster (and easier) development
  - eXtreme programming becomes natural
  - You get for free extendibility (and flexibility)
- ▷ **... with dynamic loading of compiled code for performance**
  - Can be dangerous (compatibility of compilers)
  - Can be tedious (exposing the compiled interfaces to the scripting language)
  - You can use some existing tool (**SWIG** has been proven quite good in our experience); in the end everything work smoothly

- **Find a good graphics widgets library**
  - Should be easy to use and easy to extend
  - As much as possible no dependencies from other parts of the code
  - Multiplatform and really free?
- **Use of industry accepted technology ...**
  - OpenGL
  - CORBA
  - PostScript
  - XML
- **... with a bit of new frontiers**
- **Frameworks, not applications**
  - As much as possible try for a modular and flexible structure
  - Prefer **abstract interfaces** approach or **well defined protocols** to ad-hoc solutions

Roadmap

- The long-term aim is to be able to augment the GLAST software experience by providing the same functionalities of the actual GUI, but with much more interactivity capabilities.
- We have completed (during summer) the transition to a better internal scene graph (essentially HepRep itself): FRED is now much faster
- We are checking (and continue to check) for evident compilation and runtime bugs both on Linux and Windows (we need to finalize, for example, the compiler to use under Linux; we are waiting the full transition of GLAST to gcc 3.2)
- For next release 1.0 we are planning few new features and much more work and efforts on making FRED more stable and easy to use, advertise it and start to collect user feedbacks

http://www.fisica.uniud.it/~riccardo/research/fred/
http://www.fisica.uniud.it/~riccardo/research/fred/HepRepHowTo.html



Instance of type:Particle
DrawAs: Line
Ei: 1000.000000
Eo: 0.000000
Layer: mc
Proc: primary
PDG: 22
Name: gamma
Color: white
Sel: True