



Event Display using ROOT - Experience with the BFEM and potential for extension to the full LAT

N. Lumb and G. Spandre
I.N.F.N. - Pisa

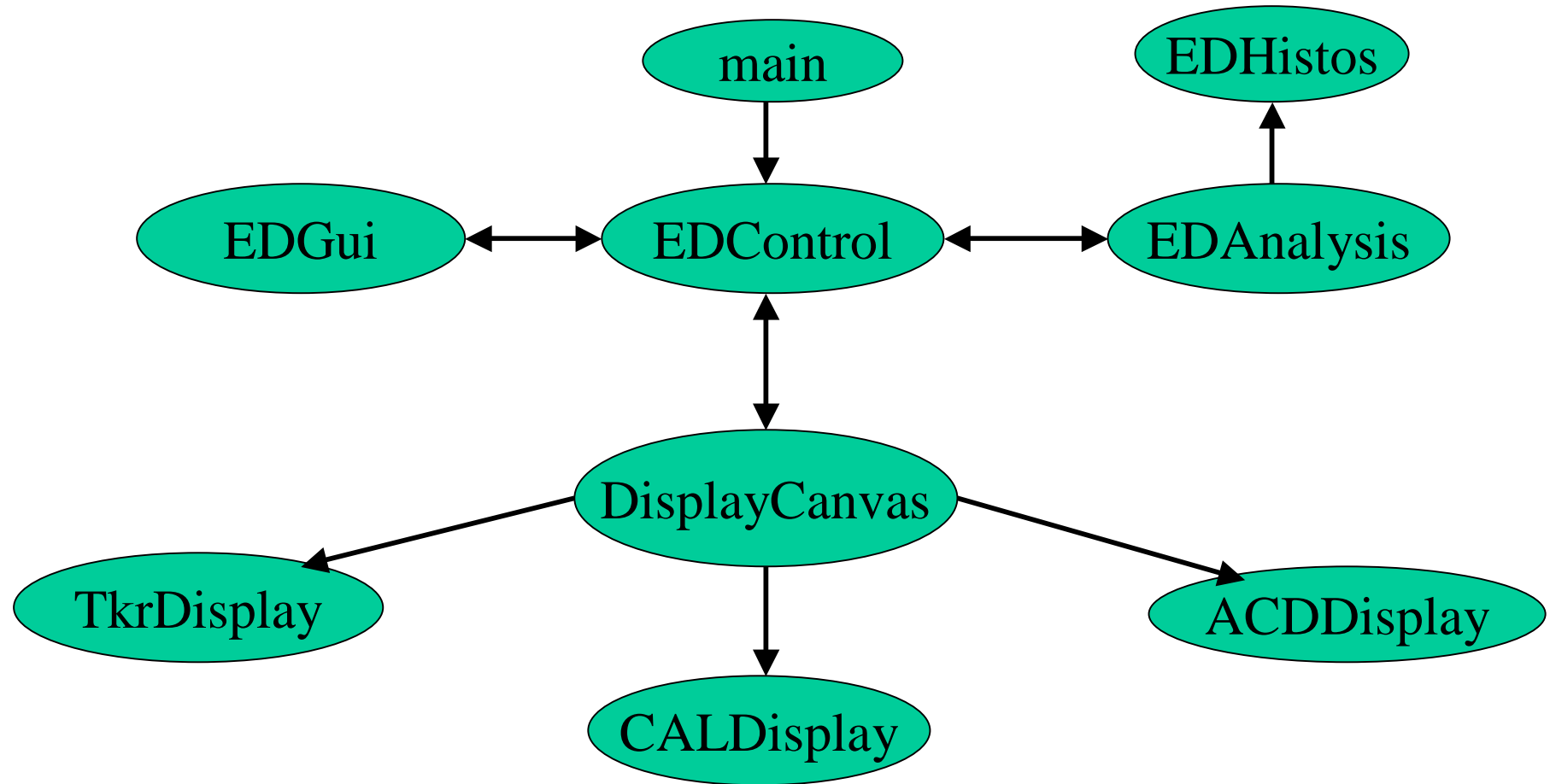


Latest developments

- bfemDisplay – fully compiled package now under CMT.
- Re-design of GUI, e.g. simpler event selection.
- Interactive setting of thresholds for colour-coded display of CAL, ACD and XGT hits.
- Improved file selection dialog.
- Handling of G4Sim files.
- HTML documentation of all classes.

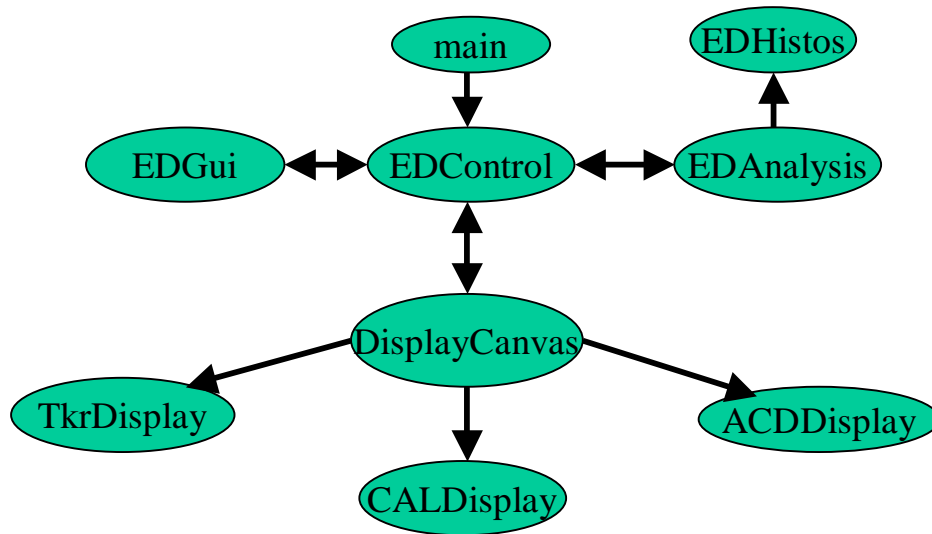


bfemDisplay Program Structure





bfemDisplay main classes



Main: ROOT set-up, starts overall control and Cint session (TRint).

EDControl: file management (TFile), event control, response to GUI commands, overall control of display (using TCanvas) and analysis.

EDGui: Draws GUI using ROOT GUI classes (TFrame, TGButton, TGMMenuBar, etc.) Responds to GUI events via ROOT ProcessMessage method.

DisplayCanvas: gets one event from file and displays on the canvas. Controls display sub-classes (TkrDisplay, etc.) Intensive use of TShape and TNode.

EDAnalysis: Analysis of raw data and display via EDHistos class.



Graphical user interface (EDGui class)

TGMenuBar
(Now just program exit)

TGButton
(Go up one directory level)

TGComboBox
(Switches file type)

TGTab
(Switches between Display and Analysis modes)

TGFileContainer
(Displays files in current directory after filtering. Highlights currently selected file)

TGButton
(Step forward/backward through file; loop over events)

TGTextEntry
(Go directly to event by specifying event no. or ID)

TGSlider
(Rough selection of position in file. Doubles as progress bar)

TGMainFrame

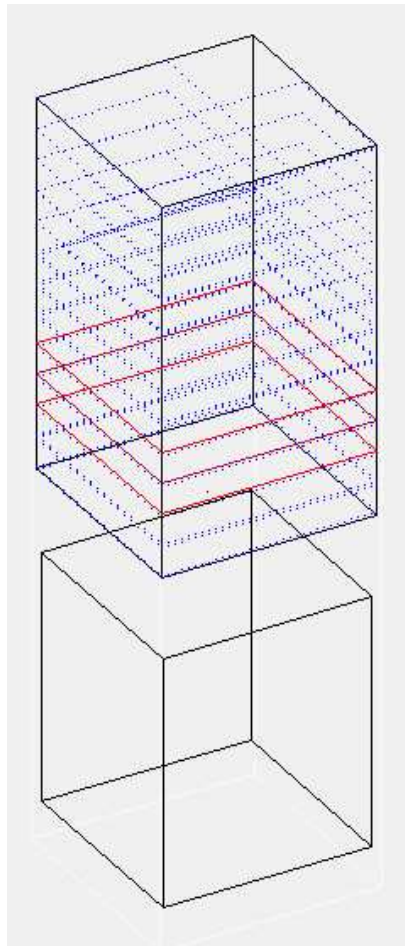
TGGroupFrame
(To set thresholds for hit display)

Thresholds (MIP)			
	CAL	ACD	XGT
Lev1	0.1	0.1	0.5
Lev2	0.5	0.5	1.0
Lev3	2.0	2.0	2.0
Lev4	10.0	10.0	4.0

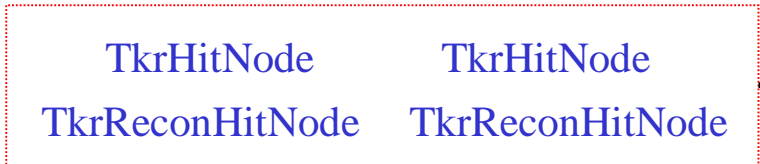
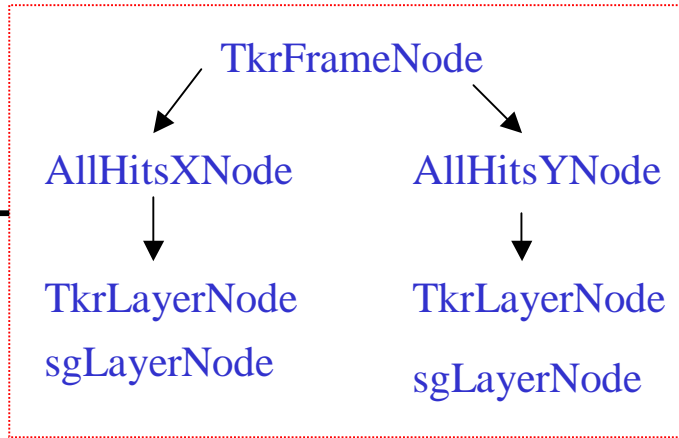
Run no. 5300 Evt ID 2793 Evt no. 270



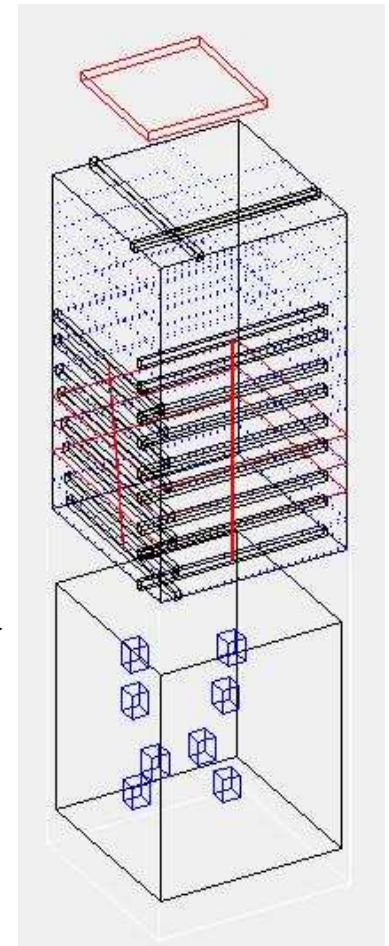
Drawing node structure (DisplayCanvas class)



Set up nodes for detector outline once only



Add raw and recon hit nodes event by event (delete old nodes!)

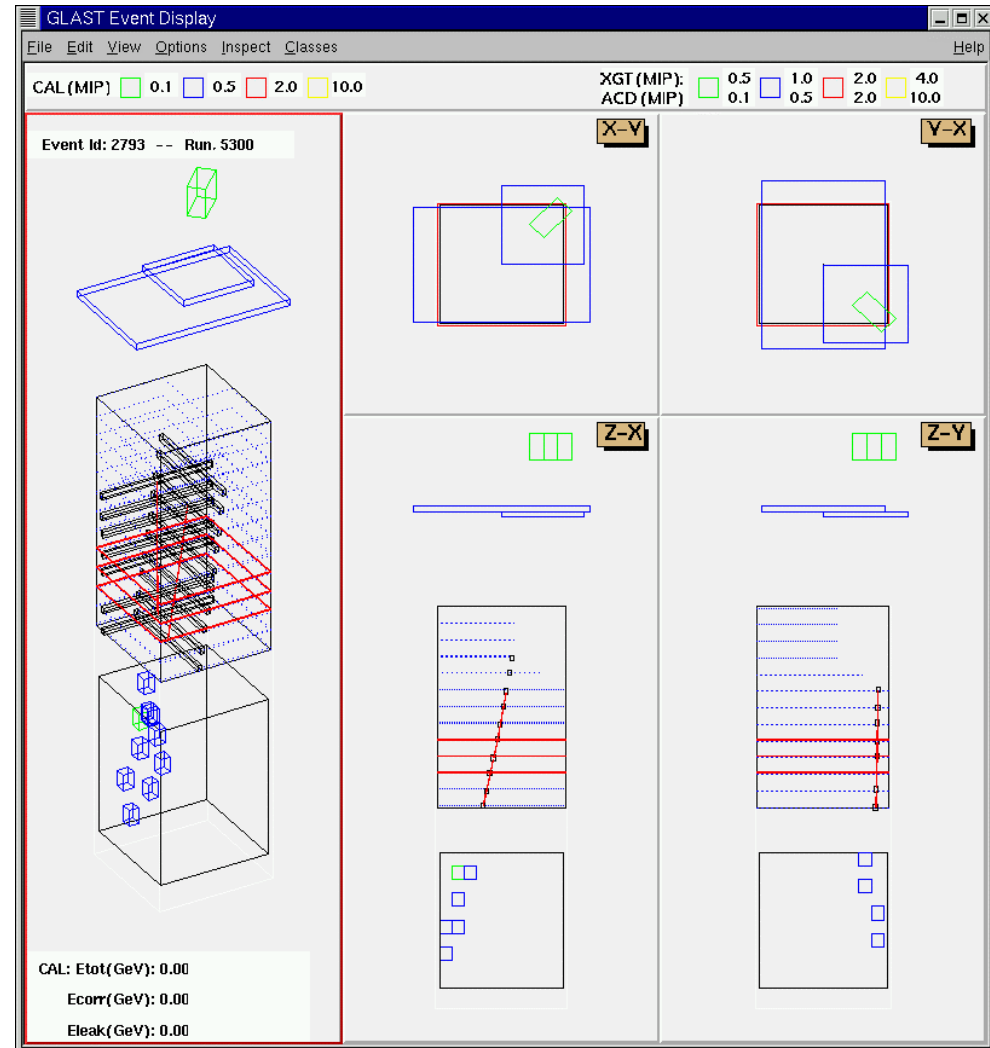


Exploit TNode::SetVisibility() to hide nodes.



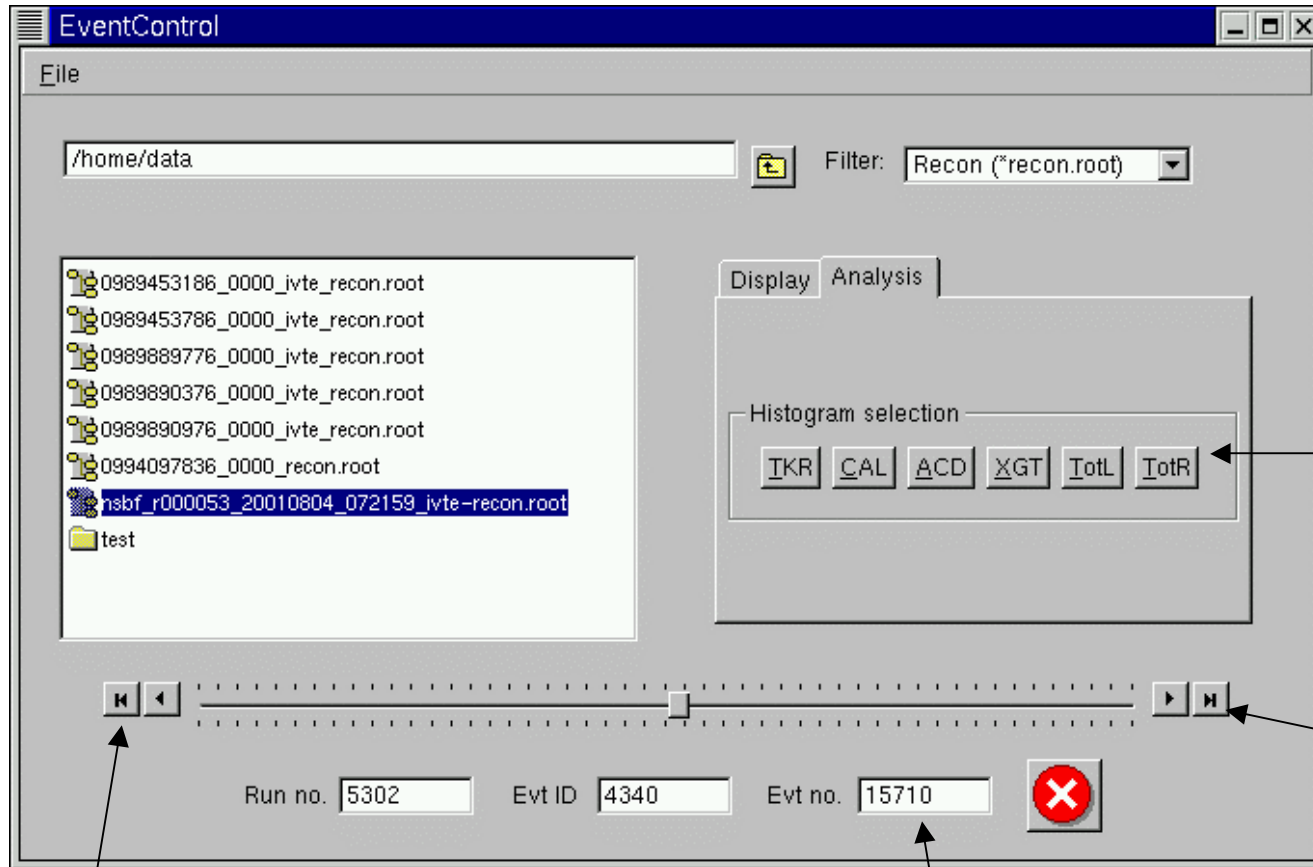
Event display drawing canvas

- Displays tkr layers – and shows position of missing ladders.
- Indicates position of SuperGlast absorber layers.
- Colour-coded energy scale for CAL, ACD and XGT.
- Many useful built-in features –
 - Interactive adjustment of titles, fonts, styles, etc.
 - Addition of labels, shapes, arrows...
 - Postscript file generation.
 - Macro generation.
 - Display with X3D, OpenGL





Options in Analysis mode (EDAnalysis class)



Histogram selection buttons

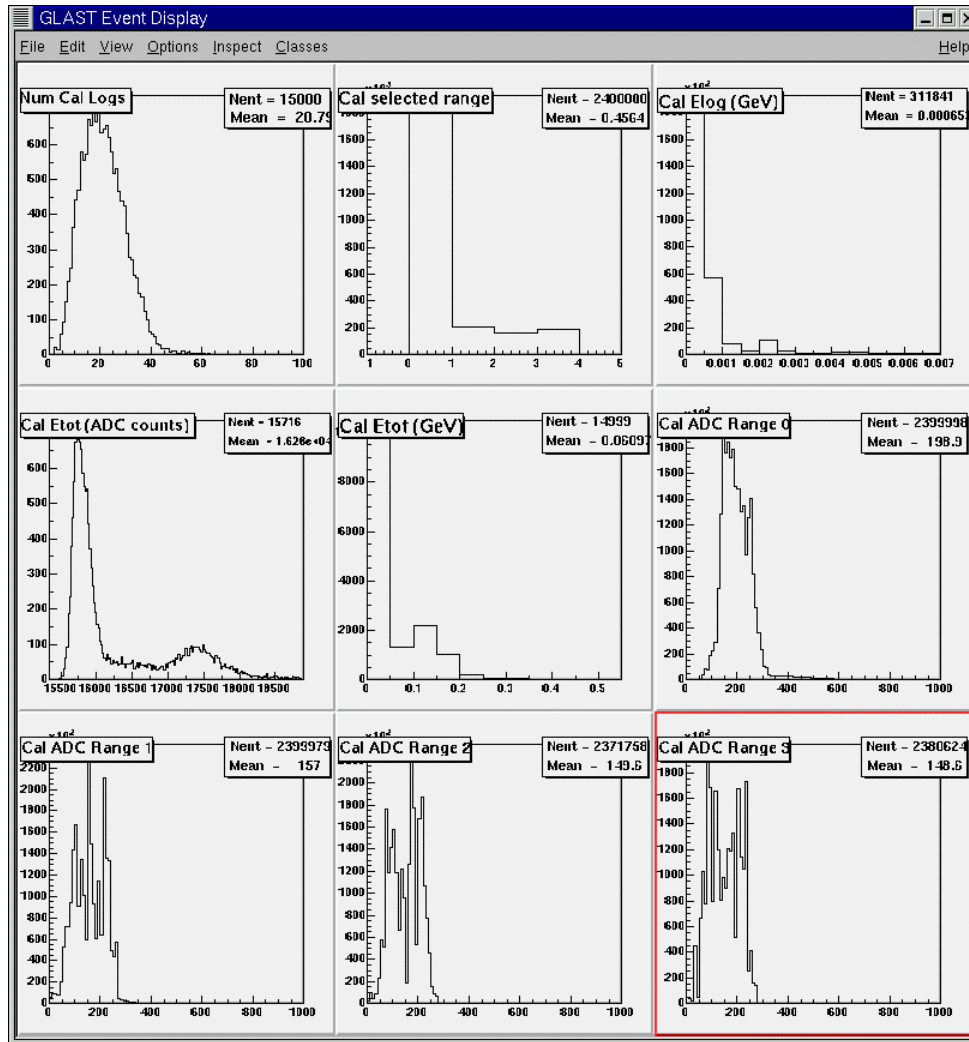
Go forward

Rewind, histo reset

Analyse up to this event (can also use slider widget)



Example of histogram display (CAL)



- Some additional histos not accessible from GUI.
- ALL histos stored in file Histograms.root
- Can be accessed either from bfemDisplay Cint session or independent ROOT session.
- Easy access using TBrowser class.



Future plans

- Fix remaining memory leaks.
- Goto event by specifying event ID.
- Implement CAL recon.
- Support of XML detector geometry files.
- More sophisticated analysis and histogramming routines?
 - Interactive thresholds active on analysis.
 - Handle recon and MC truth, housekeeping?
- Port to Windows.



Positive features of ROOT (I)

- Already widely used within the *GLAST* collaboration
 - Raw and Recon data available as ROOT files.
 - Methods to extract information from data structures already developed for analysis programs.
- TNode class provides a simple and natural means of building detector geometry and of placing hits within that geometry.
- TShape class offers all the basic building-blocks that we are likely to need to represent the detector and raw/reconstructed events.



Positive features of ROOT (II)

- Graphical displays already contain much of the basic functionality that we need, without much additional coding:
 - Rotate and zoom;
 - Click and inspect objects on canvas;
 - Display using OpenGL, etc.
- Rich set of tools to interface to window and system managers, at least under Linux.
- Uniform language (C++) for compiled code, Cint interpreter and (therefore) macros.
- Very advanced data analysis and histogramming features based on developers' experience with PAW.



Positive features of ROOT (III)

- Under Linux, GUI classes provide all of the basic functionality needed to build a graphical user interface.
- Fast and helpful replies from lead developers in response to questions.
- ROOT is widely used - existence of a large base of experience and expertise in contact via RootTalk mailing list.
- Many other potentially useful features not yet explored by the Pisa team e.g. Networking classes.



Negative features of ROOT (I)

- Relatively poor documentation. ROOT user's guide useful only for getting started. Main source of information is the HTML documentation of the classes themselves (very good) and RootTalk digest.
- Above point results in even the simplest of requirements sometimes taking a disproportionate amount of time to implement. E.g. re-sizing of histogram titles.
- Several bugs and non-implemented methods (not documented). E.g. could not use shared memory to communicate between compiled code and Cint, even after several e-mail exchanges with developers.



Negative features of ROOT (II)

- Lower level of support for Windows than for Linux. E.g. GUI classes only now being provided.
- For compiled code and shared libraries (but not necessary for macros), WE must provide support for all GLAST platforms.
- ROOT not specifically designed with event displays in mind - GUI and graphics classes probably less developed than data analysis/histogram features.
- No true ROOT guru exists within the collaboration (at least to our knowledge).



Status of Win32 GUI classes

- Current version of bfemDisplay only runs under Linux.
- Need Win32 classes to maintain support for official GLAST platforms.
- Recent communication with ROOT developers (10-11-2001).
- Win32 GUI classes in advanced state of development – see <http://root.cern.ch/root/win32progress/Win32GUI.html>
- May be released as a download option as early as next month.



Status of Win32 GUI classes (cont.)

- ROOT team gets extra manpower end Nov. (Valery Onuchin) – will assist Bertrand Bellenot with Windows port.
- Fully operational, stable version of Win32 GUI in next major release of ROOT (version 3.03).



Conclusions

- A stable, fully-compiled Event Display for the BFEM has been developed and placed in the SLAC repository under CMT/CVS.
- It is being used!... by Alicia, Eduardo, Heather, Tune,...
- Continually responding to requests for new features/improvements.



Conclusions (cont.)

- ROOT features a rich set of classes for data manipulation, histogramming, graphical display, geometry navigation and many other functions relevant to our aim of creating an event display for GLAST.
- An event display using ROOT could be easily integrated into our existing software infrastructure.
- The relatively poor documentation is offset by a wide base of expertise available for consultation.
- ROOT is becoming a mature package but still contains many undocumented bugs and inconsistencies.
- In our opinion, there is no doubt that a full event display could be written using ROOT, e.g. as HepRep client.