



Graphics Editor in ROOT

Ilka Antcheva*, René Brun, Fons Rademakers

CERN, Geneva, Switzerland

* funded by PPARC, Swindon, UK



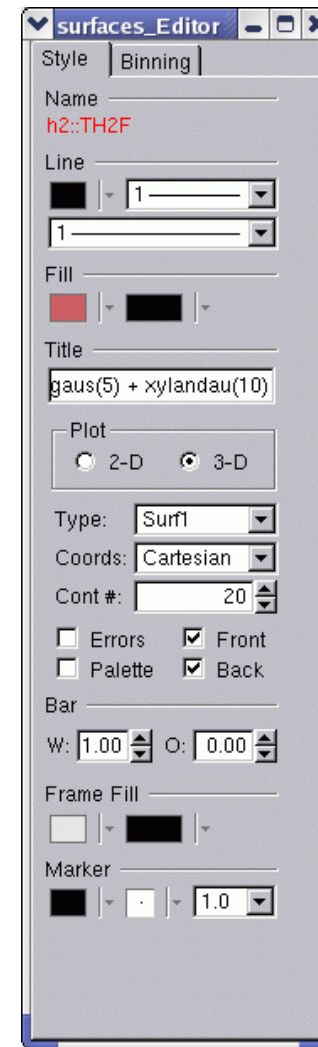
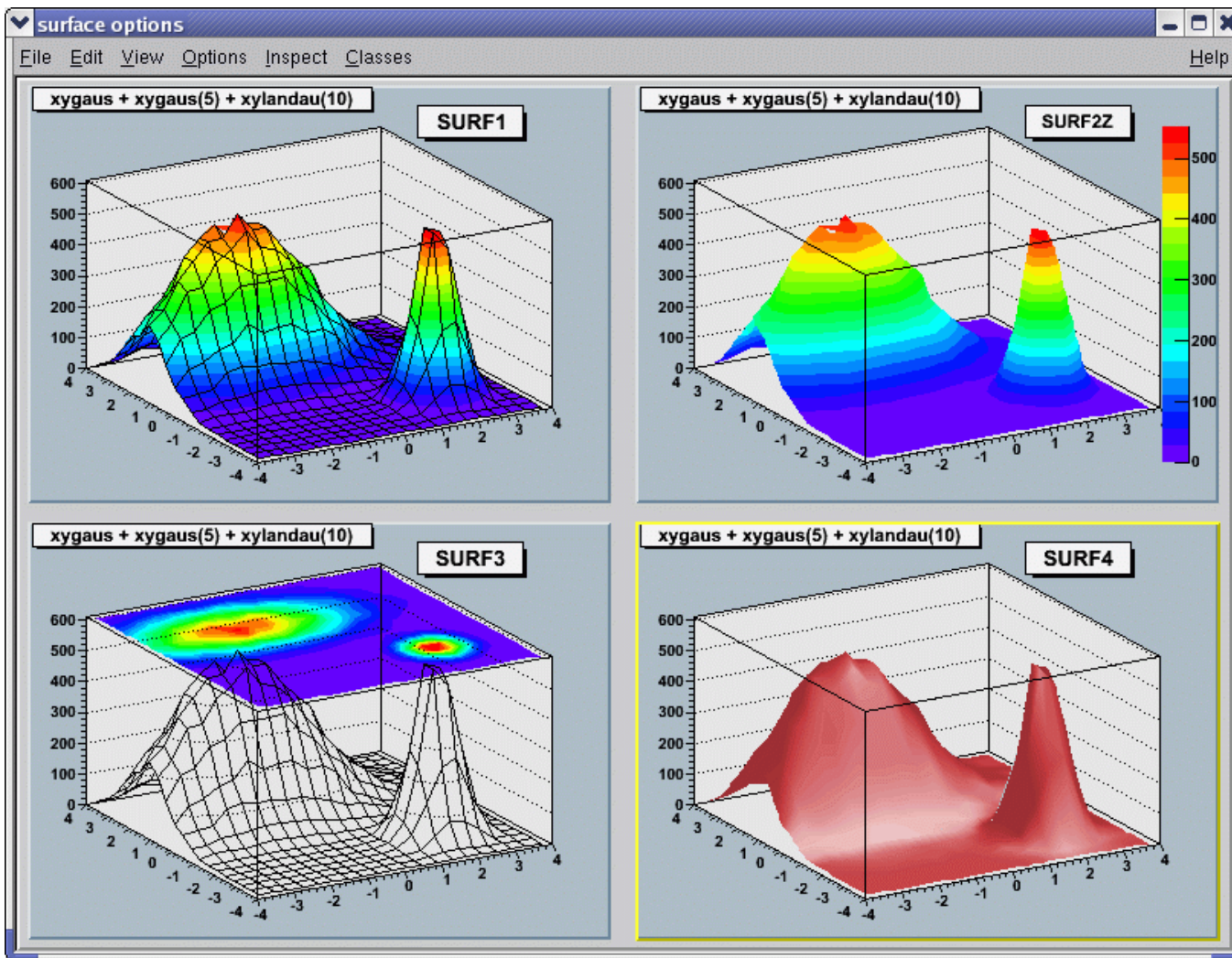


- Main Goals
- Design Solution
 - The “Look”
 - The “Feel”
 - Conceptual Elements
- Editor Usability
- Focus on Users
- Next Steps





Main Goals (1)





Main Goals (2)



- Object orientation of the editor design – keeps users focused on objects, not on how to carry out actions.
- Managing GUI complexity by splitting the graphics editor into discrete units of so-called **object editors**.
- Editor responds dynamically and presents the right GUI at the right time according to the selected object in the canvas.
- Easy-to-use.
- Require minimal user's training.
- Protect users from obvious errors.





Design Solution (1)



- Modular – it loads the corresponding object editor **objEditor** according to the selected object **obj** in the canvas respecting the class inheritance.

TArrow	TAttMarker	TCurlyArc	TH1	TPad
TAttFill	TAttText	TCurlyLine	TH2	TPaveStats
TAttLine	TAxis	TFrame	TGraph	

- Algorithm:
 - Search for a class name **objEditor** (correct naming is important).
 - Check that this class derives **TGedFrame** (the editor base class).
 - Make an instance of the object editor using **TROOT::ProcessLine** method.
 - Scan all base classes for corresponding object editors.

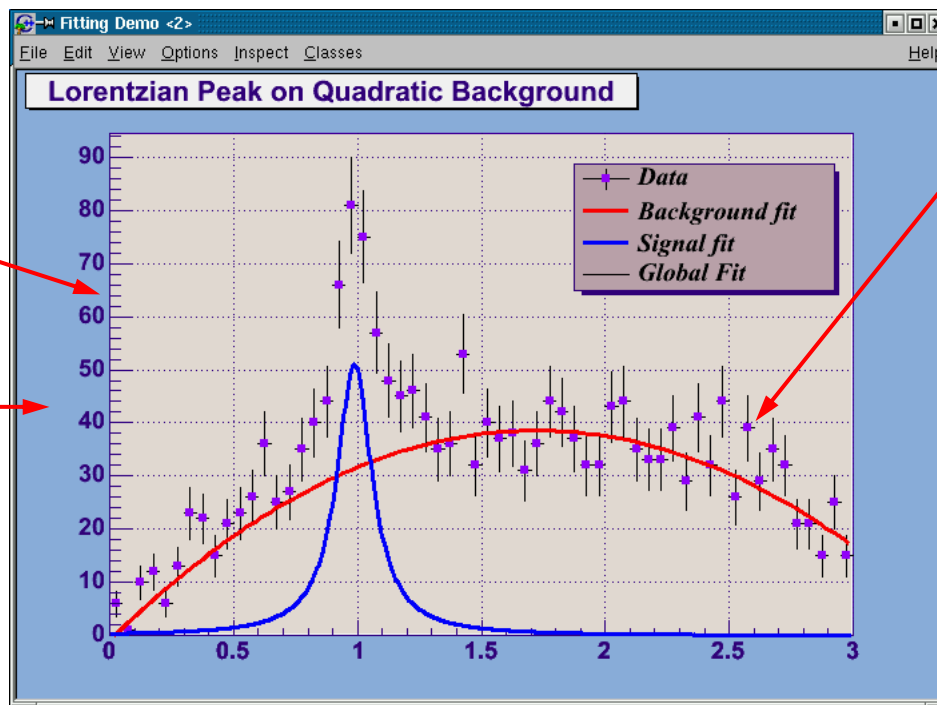
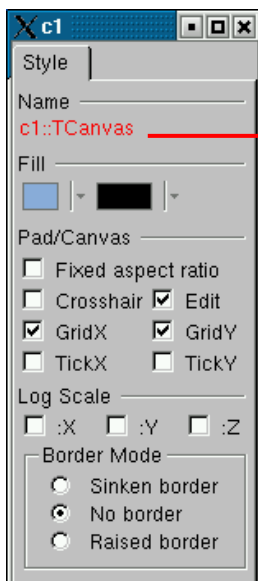
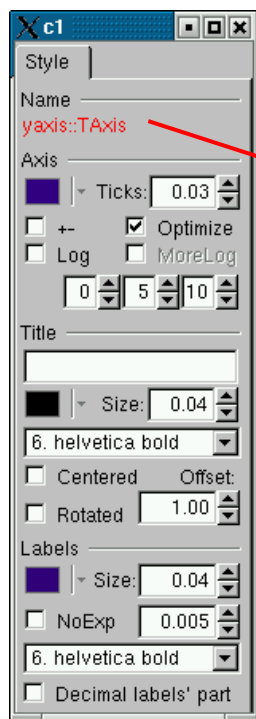




Design Solution (2)



- Signals/slots communication mechanism handles GUI actions.
 - Canvas sends a signal which object is selected.
 - Corresponding object editor is activated and ready for use.

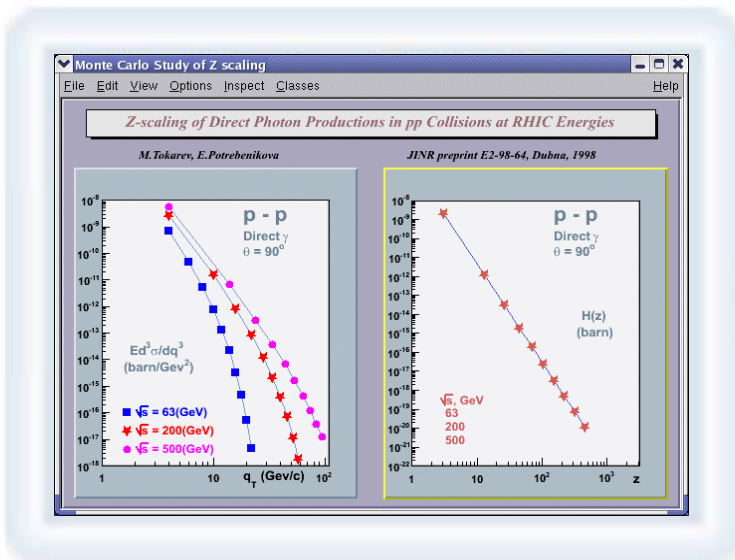
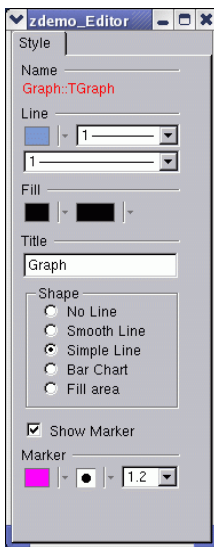
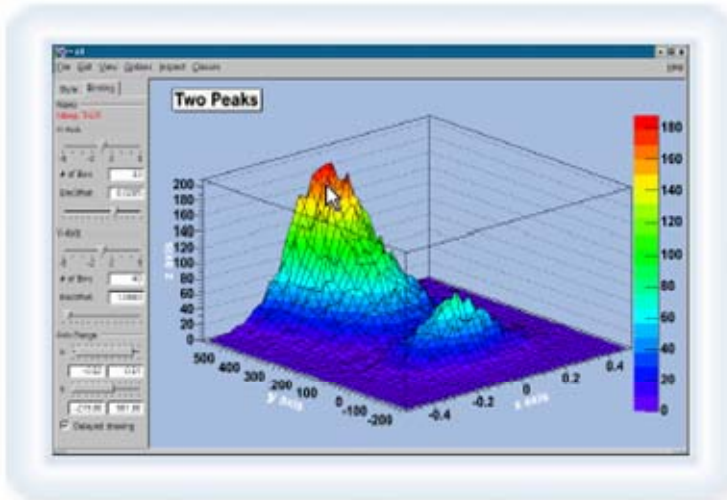




Design Solution (3)

ROOT graphics editor can be:

- Embedded – connected only with the canvas in the application window
- Global – has own application window and can be connected to any created canvas in a ROOT session.

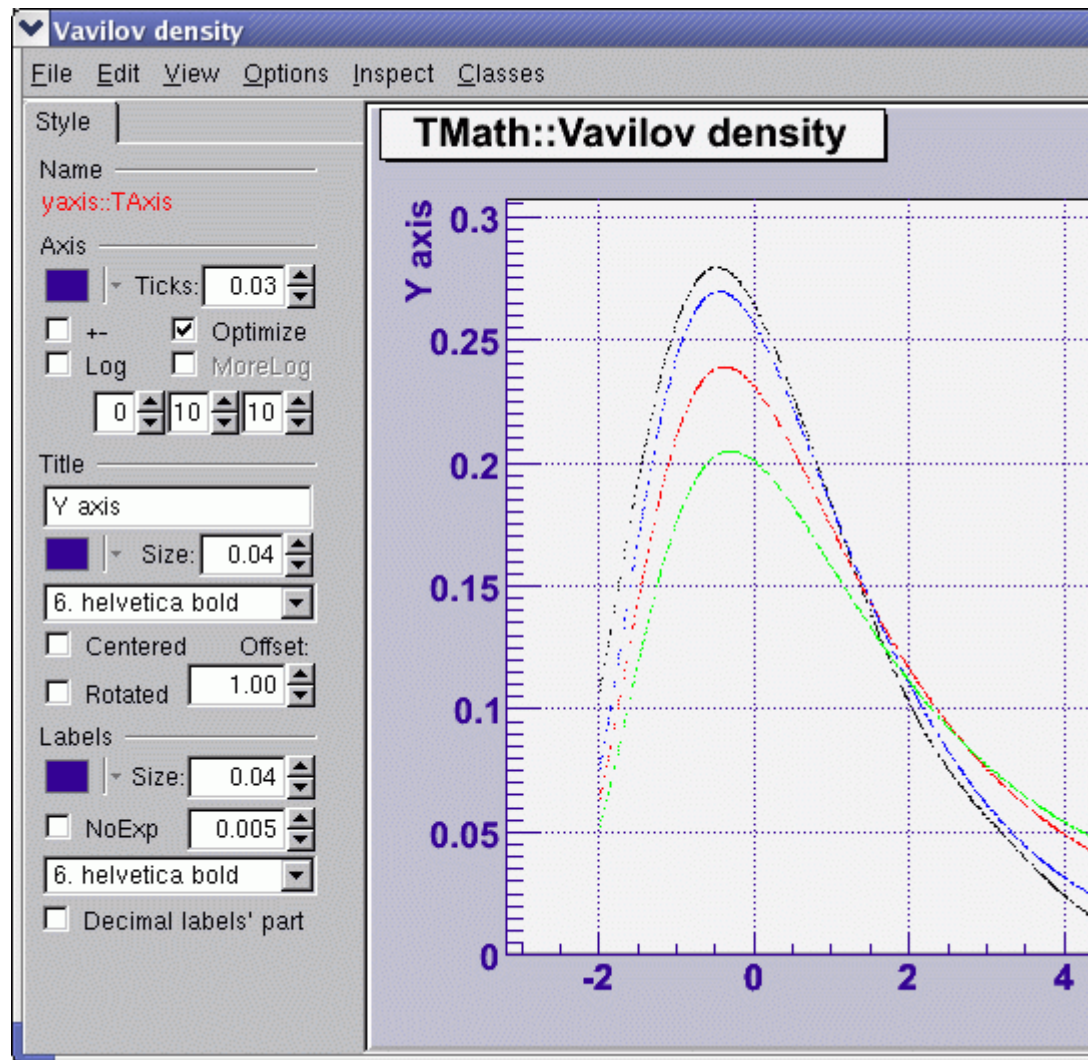




The “Look”



- Color choice to point user’s attention to selected object.
- Font choices – by default.
- GUI elements: buttons, combo boxes, number and text entries, etc.
- Overall visual impression - layout shows the visual hierarchy, consistency and balance.
- Additional redundant texts and labels are avoided by grouping elements.
- Tooltip based help.

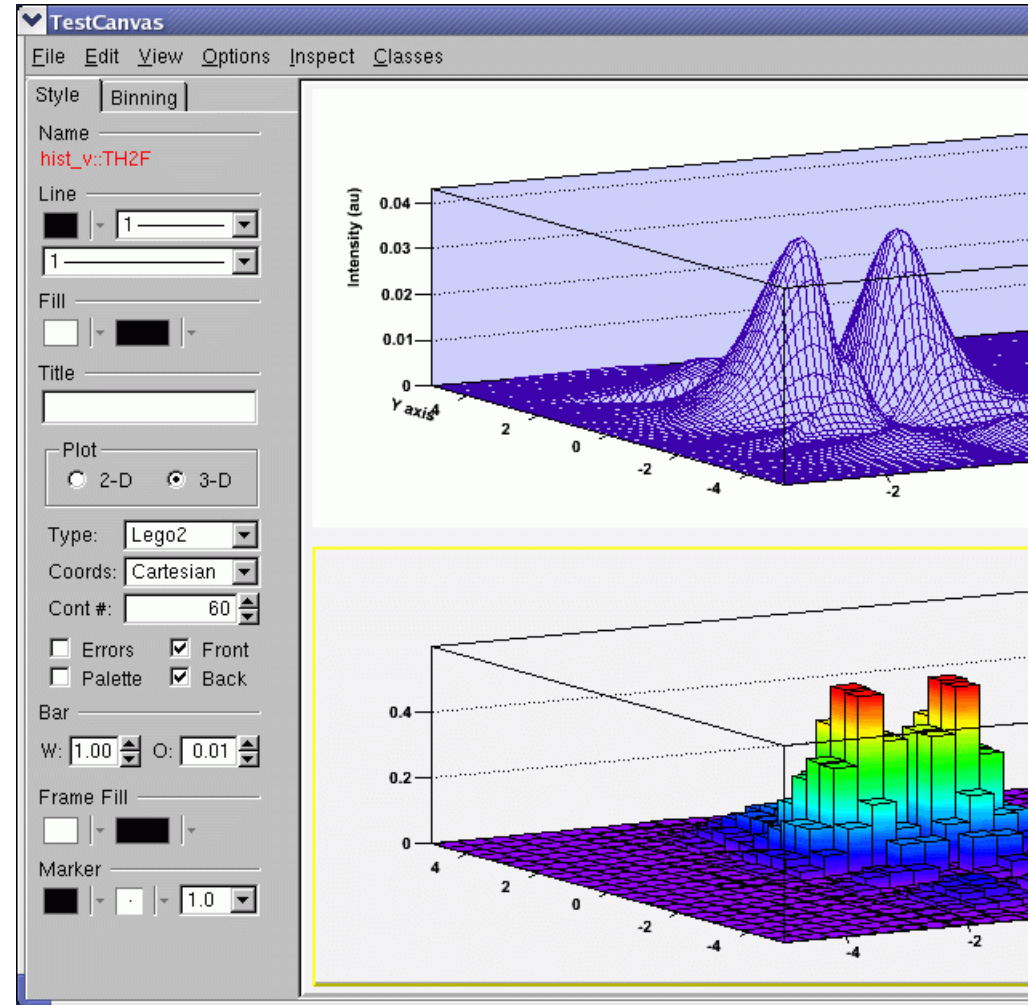




Conceptual Elements (1)



- Easy-to-use.
- Recognition over recall determines design choices.
- Power of each GUI action – same things work same way; full and continuous feedback on the action result.
- Flexibility for change – only information relative to the current task is presented; other GUI parts are hidden.
- Capacity for growth.



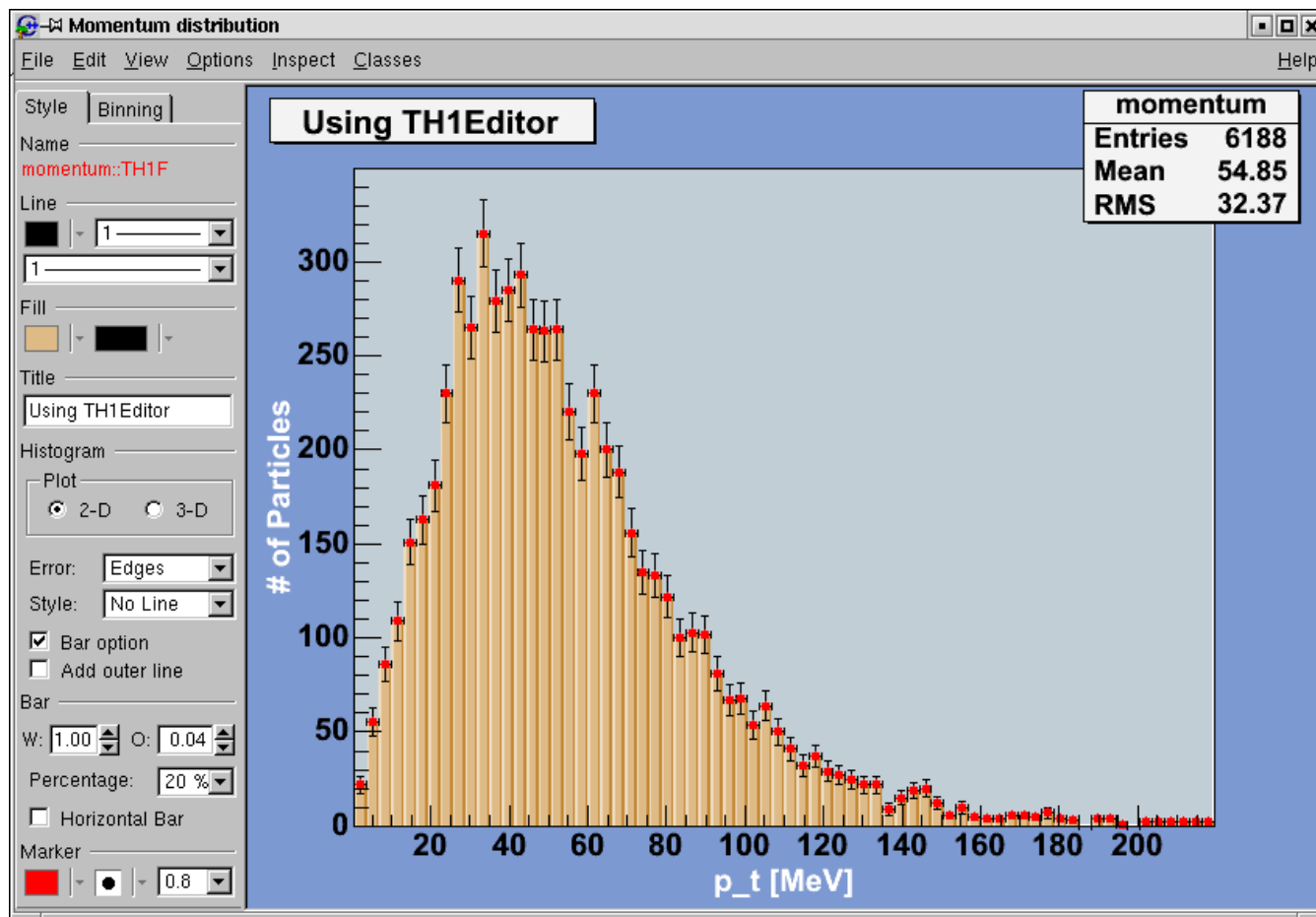


- Can be extended easily by any user-defined object editor - this makes GUI design easier and adaptive to the users' profiles.
- Rules to follow:
 - Derive in the code from the base editor class **TGedFrame**.
 - Correct naming convention: the name of the object editor should be the object class name + 'Editor'.
 - Register the new object editor in the list **TClass::fClassEditors** in the end of its constructor.
 - Use signals/slots communication mechanism for event processing.
 - Implement **SetModel** method where to set GUI widgets according to the object's attributes.
 - Implement all necessary slots and connect them to appropriate widget signals.





- To reduce...
 - Visual work
 - Movements
 - Intellectual work
 - Memory work
- Benefits
 - Higher productivity
 - Overall validation
 - Users' confidence



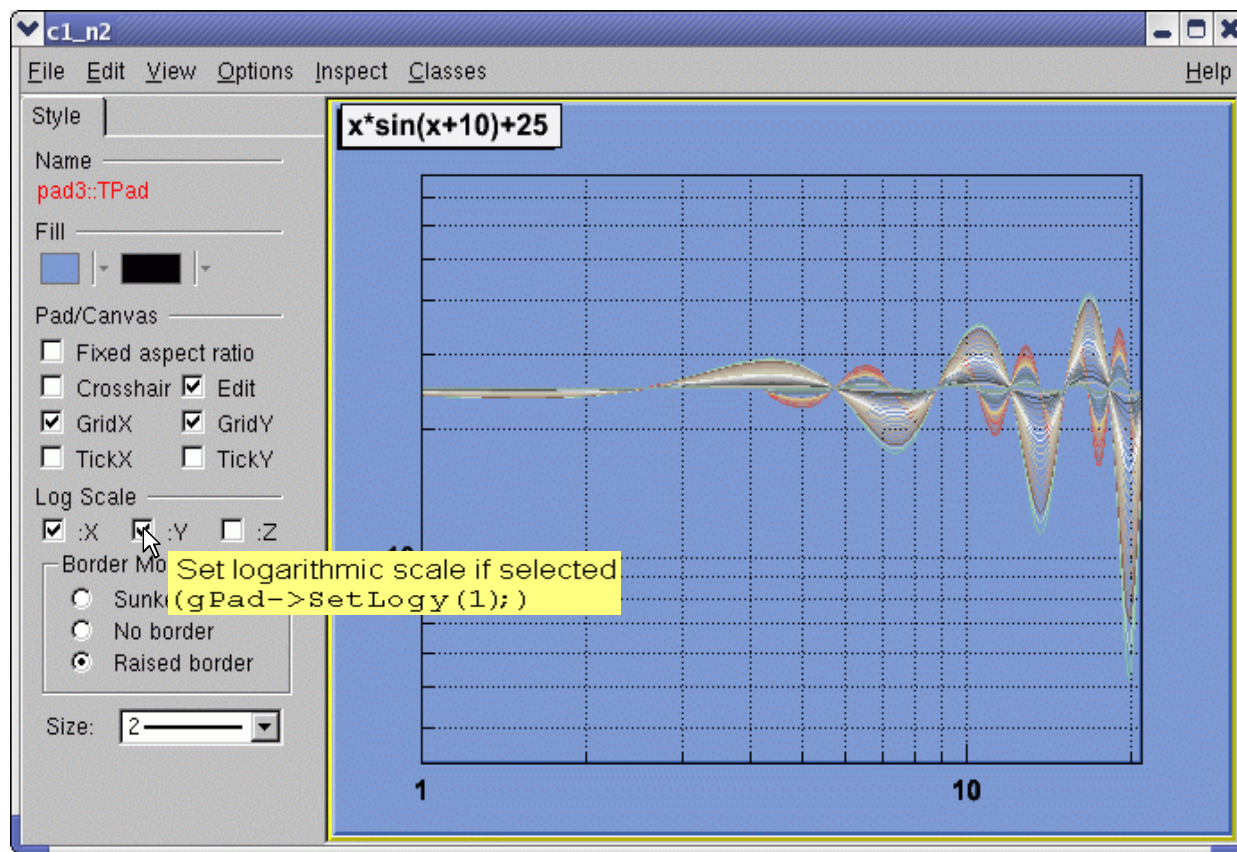


- **Novices** (*for a short time*)
 - Theoretical understanding, no practical experience with ROOT
 - Impatient with learning concepts; patient with performing tasks
- **Advanced beginners** (*many people remain at this level*)
 - Focus on a few tasks and learn more on a need-to-do basis
 - Perform several given tasks well
- **Competent performers** (*fewer than previous class*)
 - Know and perform complex tasks that require coordinated actions
 - Interested in solving problems and tracking down errors
- **Experts** (*identified by others*)
 - Ability to find solution in complex functionality
 - Interested in theories behind the design
 - Interested in interacting with other expert systems





- To include ROOT commands in tooltips of the GUI widgets
- Hide/Show objects
- New object editors
- Help
- HowTo design object editors
- Undo/Redo
- Style Manager
- Fit Panel GUI



```
root[9] gPad->SetLogy(1);
```

