Timo Taglieber

The GTK+ Project

Implementation of a new visualization frontend for MUSCOD-II using GTK+ and Cairo

Software Practical
Summer Semester 2008
Interdisciplinary Center for Scientific Computing
University of Heidelberg

Content

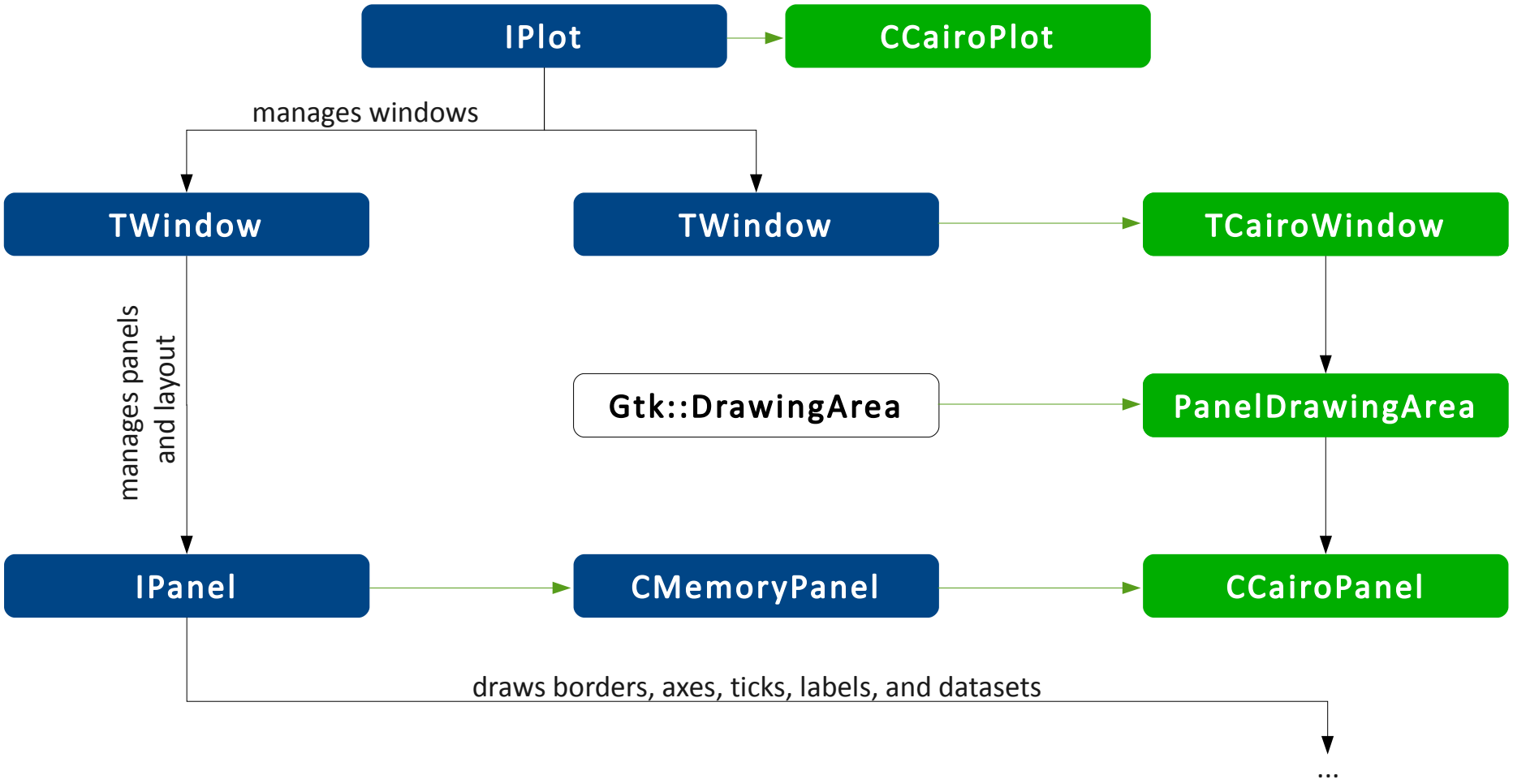
The Task

- PLOT: visualization frontend for MUSCOD-II written in C++
- Output of iterative algorithms can be plotted to several media (e.g. Screen, Postscript)
- PLOT uses the PGPLOT Graphics Subroutine Library, which..
  - is written in Fortran-77
  - is only free for non-commercial use (not public domain)
  - was last updated in 2001
  - has limited features (e.g. only 16 colors, only float precision)
- Task: Re-implementing the PLOT interface with Gtk+ and Cairo
  - Using up-to-date free software with C++-bindings, more flexibility, extensibility, output formats, etc. (Firefox 3 renders everything with Cairo!)

Implementation of a new visualization frontend for MUSCOD-II using GTK+ and Cairo

The PLOT interface

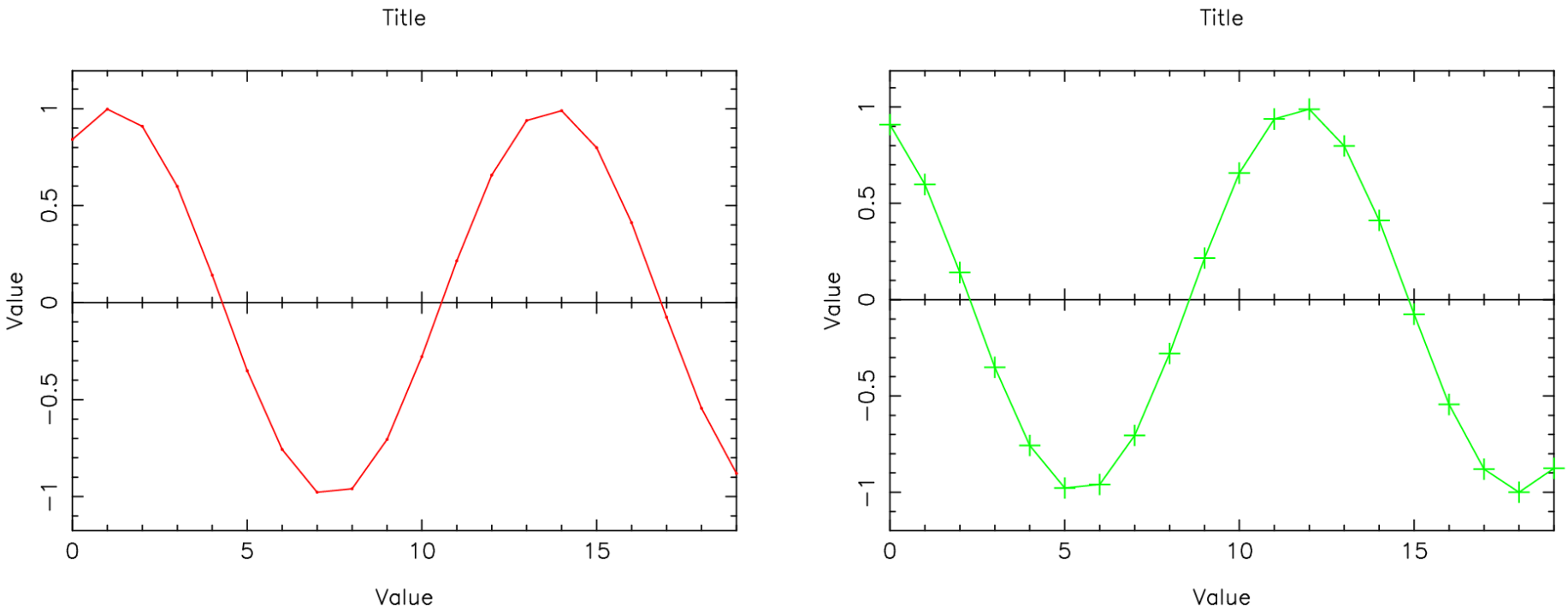
- **Inheritance** and **interaction** of involved classes



Implementation of a new visualization frontend for MUSCOD-II using GTK+ and Cairo

Existing implementation (using PGPLOT)

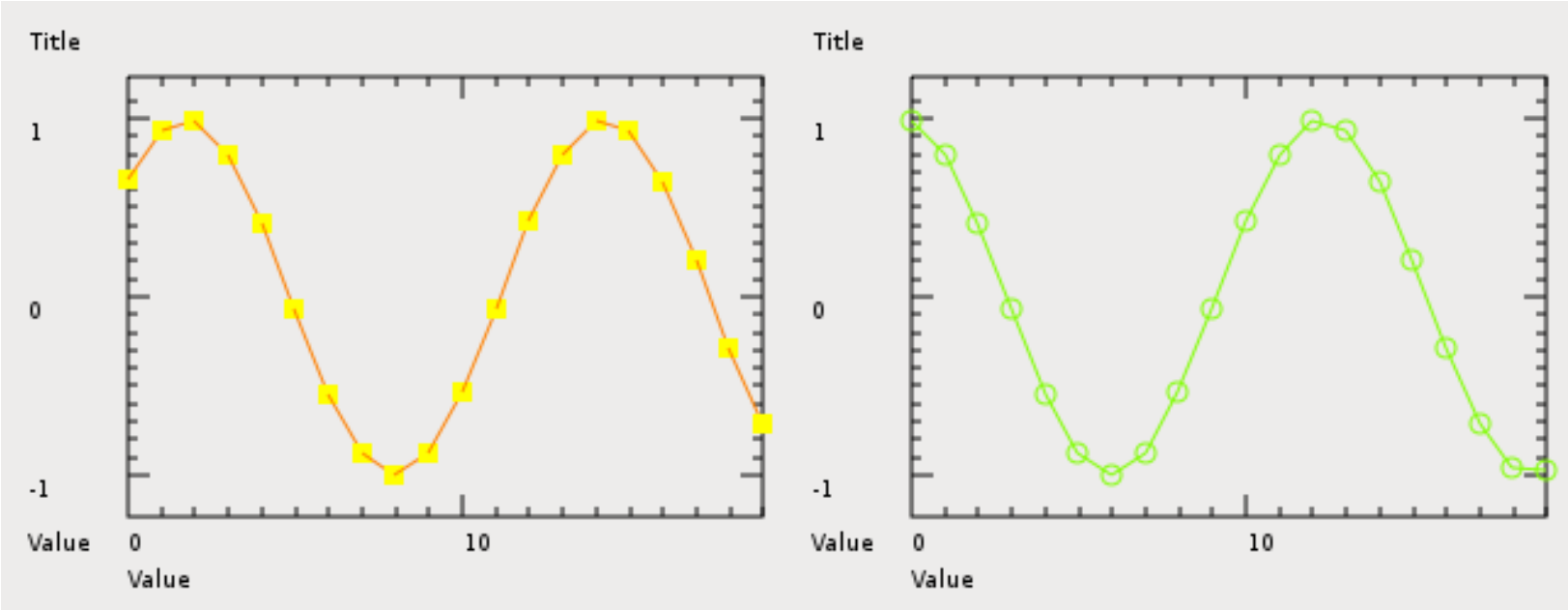
- Plotting two panels using PGPLOT
- ...



Implementation of a new visualization frontend for MUSCOD-II using GTK+ and Cairo

New implementation (CairoPlot)

- Built upon PLOT interface
- Colors, markers, and line styles from PGPLOT are emulated
- Panel width and height is scalable



Implementation details

- Optional autoscale mode for lower and upper bounds
- Panels are placed within table grid
- Window title shows number of visible panels
- Drawing is done with geometric primitives (lines, arcs, ...) except for the rendering of text (Pango)
- Datasets are plotted by transforming x/y values to pixels
- Code is comprehensively commented for use with doxygen (HTML/Tex/PDF export)

Evaluation of the results

- So far only screen output
- Emulation of line and axis styles to be completed
- Rendering of text troublesome: alignment and rotation
- Code covers all required functions, is well structured and „just works“
  - however not written by a C++-Guru
- Future Work:
  - Interactive features (zoom, panel order, colors, etc.)
  - dataset history related features (visualize iterative algorithm)
  - Include more output formats: PDF, SVG, PNG

Personal gain & experience

- First C++ GUI application i've written so far
  - First GUI application ever written
    - First larger C++ application written so far
- IDE handling can be troublesome (Eclipse Platform)
- Makefile basics (vs. Java-ANT)
- Working with API documentation of Gtk+, Cairo and Pango
- First impression of widget-interaction in Gnome

Sources

- Gtkmm: C++ Interfaces for GTK+ and GNOME
  `http://www.gtkmm.org`
- Cairomm: A C++ wrapper for the cairo graphics library
  `http://www.cairographics.org/documentation/cairomm/`
- Using Pango with Cairo
  `http://library.gnome.org/devel/pango/unstable/pango-Cairo-Rendering.html`
- Doxygen: Source code documentation generator tool
  `http://www.stack.nl/~dimitri/doxygen/`
- C++ Reference
  `http://www.cppreference.com`
- PGPLOT Graphics Subroutine Library
  `http://www.astro.caltech.edu/~tjp/pgplot/`