GTK+

Owen Taylor
Red Hat, Inc.

February 14, 2001

History

- Created for the GIMP (GNU Image Manipulation Program)
- Version 1.0 released early 1998
- Version 1.2 released early 1999
  - Added themes, DND, new widgets
- Version 2.0 currently in beta – forms core of GNOME–2.0 platform.

The Plus

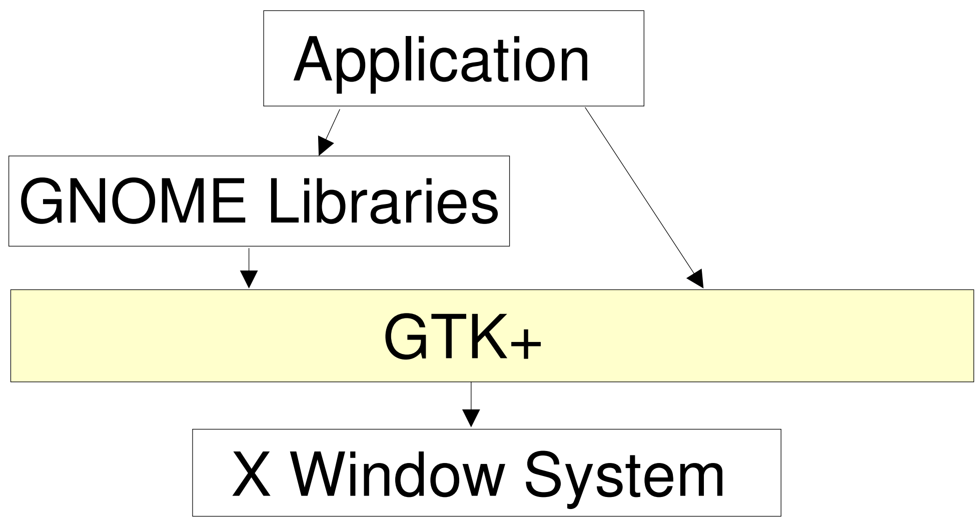
- GTK+ Derived from short–lived earlier toolkit GTK (never publically released)
- Not pronounced

GTK+–2.0

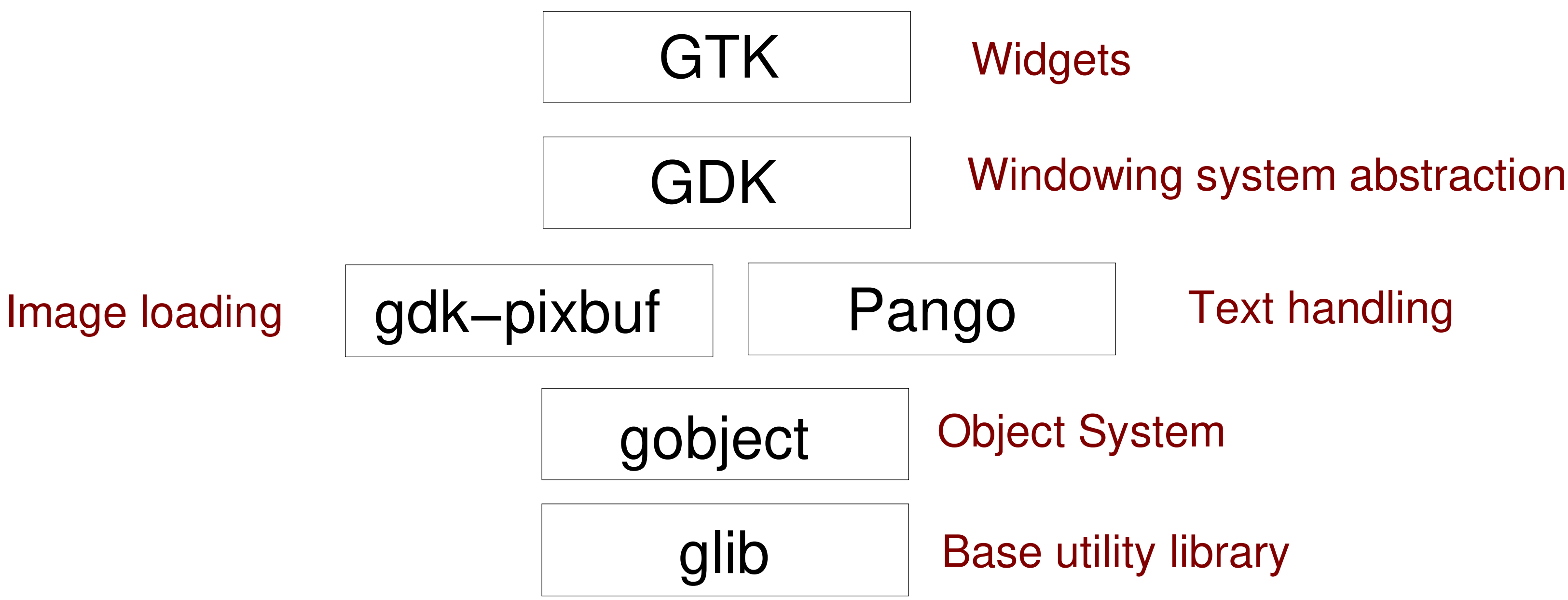
- Greatly enhanced internationalization:
  - Pango, Unicode, new Input Method support
- New Tree, Text widgets
- gdk–pixbuf – image loading
- Core object system separated out – GObject
- Many API improvements – easier programming

Where GTK+ fits in

- GTK+ provides the widget toolkit for the GNOME desktop
- Also useful for standalone, and even embedded apps.



The GTK+ Libraries



GLib

- Data structures: Hash tables, Linked list, arrays...
- Unicode manipulation functions
- "XML–like" parser GMarkup
- Utility functions for starting processes, memory management, ...
- Cross platform thread abstraction

The GLib Event Loop

- Used for timeouts, IO events, GUI events.
- Extensible with new types of events (GUI events from GDK done this way)
- For GLib–2.0, can run multiple main loops in separate threads.

GObject

- Provides object oriented programming
  - In C
  - Glue to other programming languages
- Single inheritance, interfaces
- Allows for dynamically loaded types.
- Signals
- Properties

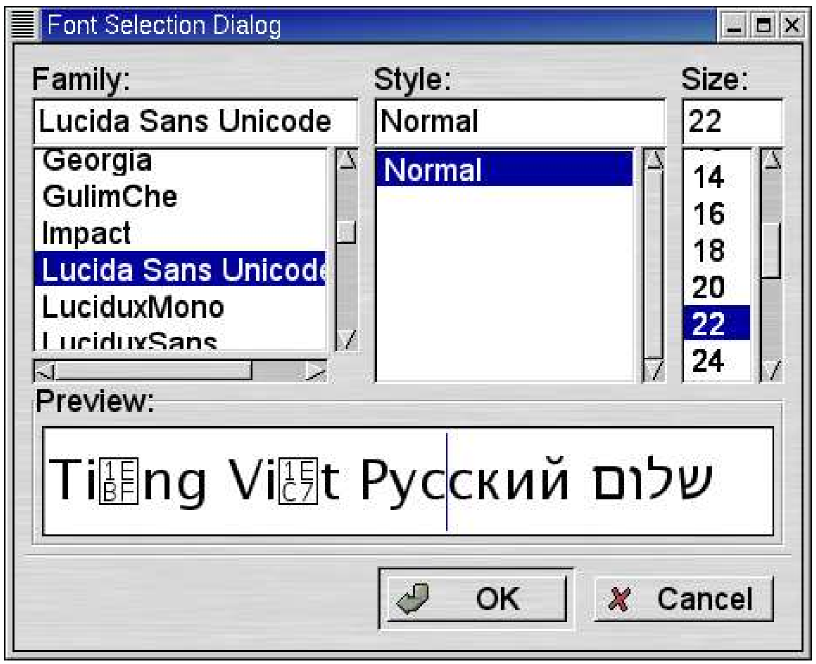
gdk–pixbuf

- Library for loading and manipulating images
- Supports many formats, PNG, JPEG, GIF, TIFF, BMP...
- Also routines for scaling and compositing images.

Pango

- Handles all text layout and display for GTK+
- Multiple font backends
  - X fonts
  - OpenType layout fonts with X render extension
  - TrueType fonts on framebuffer
- Almost all languages of the world
  - Europe, East Asia, Arabic, Hebrew, Hindi...
- Usually works behind–the–scenes.

Pango in Action



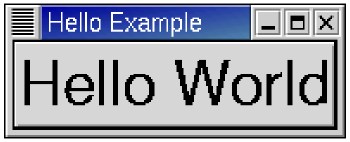
Internationalization

- All text display done through Pango
- All text in Unicode.
    - For C, strings are in UTF−8
- Supports multiple types of input methods
    - GtkIMContext input method modules
    - XIM input methods
    - IIIMF support coming

GDK –GTK Drawing Kit

- Abstraction layer between GTK+ and windowing system.
- Drawing calls
- Input events (mouse, keyboard, ...)
- Has been ported to Win32, Framebuffer.

A Simple example




```
#include <gtk/gtk.h>

static void
button_clicked (GtkWidget *widget)
{
  g_print ("I was clicked!\n");
}

int
main (int argc, char **argv)
{
  GtkWidget *window, *button;

  gtk_init (&argc, &argv);

  window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
  gtk_signal_connect (GTK_OBJECT (window), "destroy",
                      GTK_SIGNAL_FUNC (gtk_main_quit), NULL);

  button = gtk_button_new_with_label ("Hello World");
  gtk_signal_connect (GTK_OBJECT (button), "clicked",
                      GTK_SIGNAL_FUNC (button_clicked), NULL);
  gtk_container_add (GTK_CONTAINER (window), button);

  gtk_widget_show_all (window);

  gtk_main ();

  return 0;
}
```

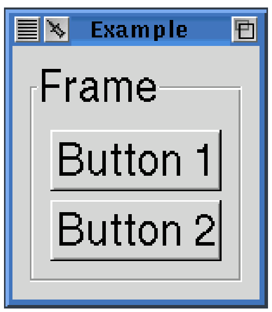
Signals

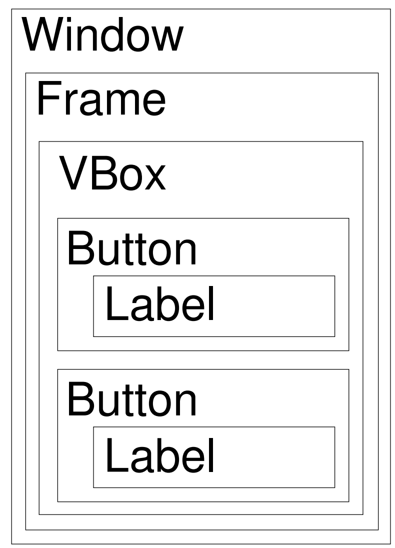
```
static void
button_clicked (GtkWidget *widget)
{
  g_print ("I was clicked!\n");
}

gtk_signal_connect (GTK_OBJECT (button), "clicked",
                    GTK_SIGNAL_FUNC (button_clicked), NULL);
```

- Establish callbacks for user interface events
- Both low–level events ("button_press_event"), and high–level events ("clicked").
- Part of GObject system, so can be used for non–GUI notification.

Geometry Management

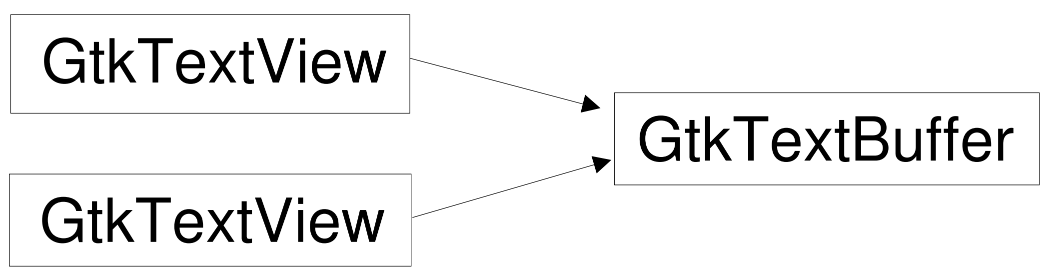
- Simple containers – GtkHBox, GtkVBox, GtkTable
- Nest to produce more complicated effects



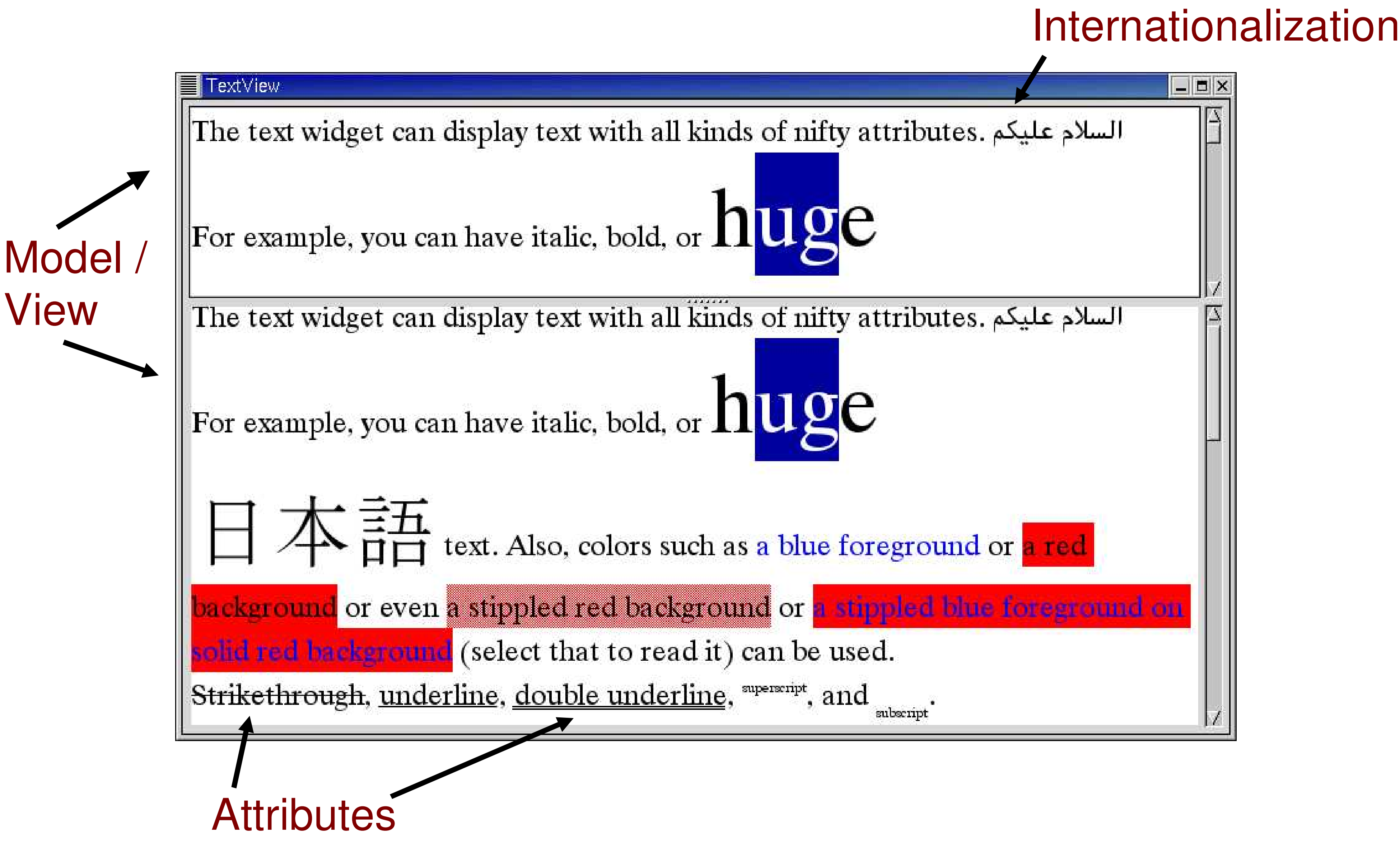


GtkTextView

- New text widget in GTK+–2.0; port of Tk text widget with many new features:
  - Model–view – one GtkTextBuffer can be used by many GtkTextViews
  - Supports full capabilities of Pango.
  - Styles, embedded pixmaps.



GtkTextView in Action



GtkTreeView

- Also new in GTK+2.0
- Used for both trees and lists
- Model−view.
  - Models implement GtkTreeModel *interface.*
  - Provided models: GtkListStore, GtkTreeStore, GtkTreeModelSort
  - Can create your own models

Properties

- Named attributes for GObjects
- Used by GUI builders
- Used from C to configure properties of widgets, GtkTextTag, etc.

```
g_object_set (GOBJECT (tag),
              "fg",      "blue",
              "weight", PANGO_WEIGHT_BOLD,
              NULL);
```

Making it easier

- Use libglade
  - Glade creates XML resource files describing GUI
  - libglade loads them into application at runtime.
- Use a scripting language (Python, Perl, Ruby, ...)
  - Greatly reduces amount of typing, compile cycle
  - Higher level programming
  - Red Hat uses gnome–python for all internal GUI development.

Resources:

- http://www.gtk.org
- http://www.pango.org
- http://developer.gnome.org (API docs)
- http://www.gtk.org/mailinglists.html
- Quite a few books, in particular:
  - GTK+ / Gnome Application Development (Havoc Pennington, New Riders)