Graduating to GUI

PyGObject for Beginners

Presented by

Paul W. Frields

Red Hat, Inc. / Fedora Project

Today's Topics

1. Getting started
2. GObject introspection
3. Classes, inheritance, hierarchy
4. Signals
5. Putting it together: Simple example

Do you know the way to GTK?

Tools

- Python
- PyGObject >= 2.28
- GTK+ >= 3.0
- Text editor of choice
- glade3
- devhelp

Getting tools

- Reasonably simple on all distributions – use your distro's package manager to easily install the proper collection
- For example, on Fedora or openSUSE: Use Add/Remove Software tool to add **gtk3-devel, gtk3-devel-docs, pygobject2**

Workflow

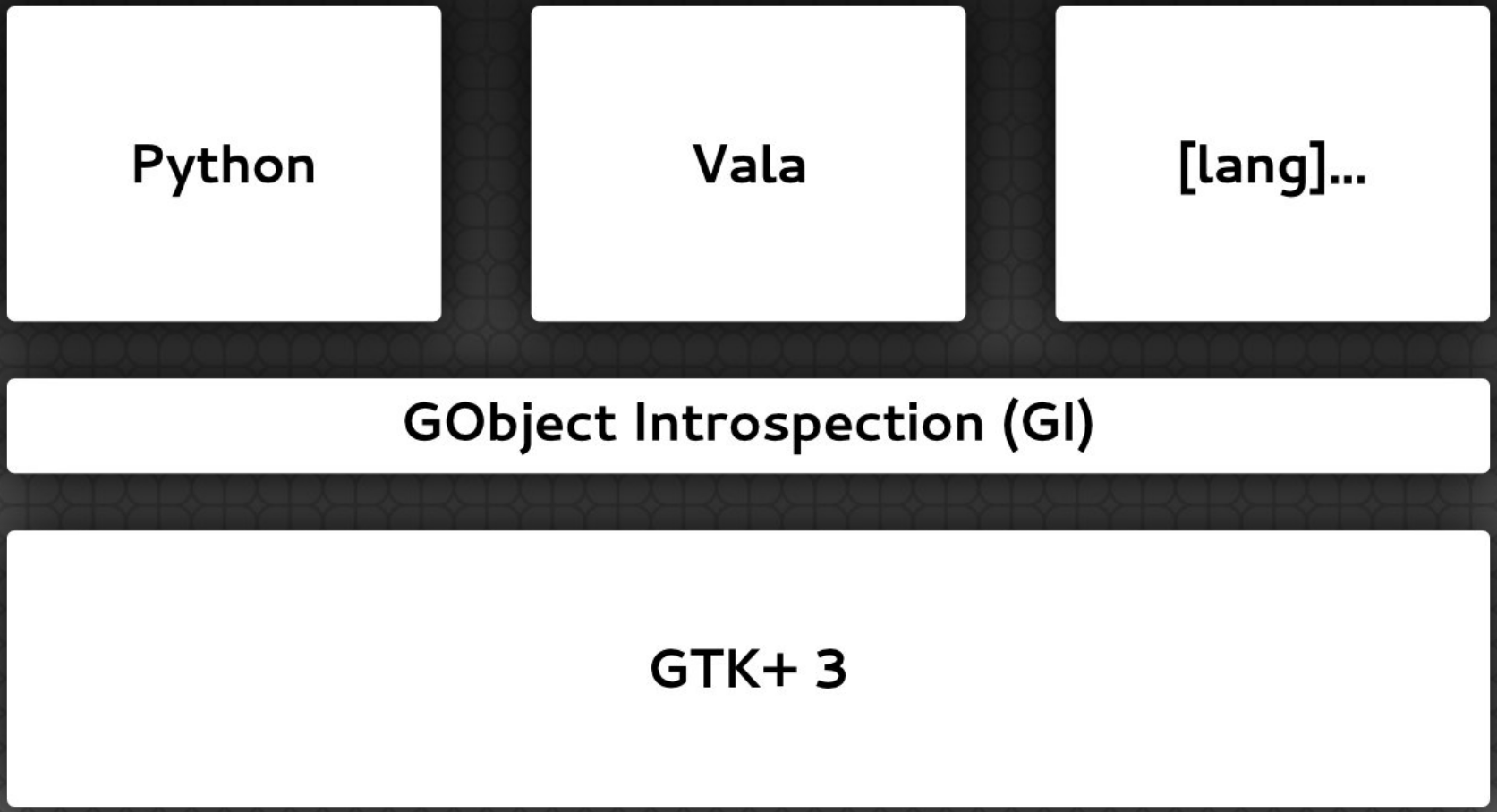
- Glade to design UI (as GtkBuilder)
  - Saved as XML file
  - Can be tweaked in Glade or any editor
- Python code loads the XML file as a resource
  - Interactive elements assigned to objects
  - Functions called based on interaction

I still have
that other gir

Who died & left you king?

- Every time GTK+ changed, PyGTK had to be updated too
- Using GObject introspection (GI), that's no longer necessary
- The GI repository (GIR) for a library makes it simple to generate bindings for many languages
- PyGTK is the old stuff, PyGObject is the new hotness and where things are going

GObject introspection



fedora

So what?

- So… well, nothing really, unless you need to port existing code – not covering that here
- Beginners should be aware PyGTK code on the intarwebz is in danger of becoming obsolete, or just plain wrong *(gasp!)*
- (See earlier version of this talk for specific PyGTK guidance, URL on last page)

A Chair() is still a Chair()

GTK object model

- Based on classes and inheritance
- Each object can have its own special properties and methods
- Real-life example: “Chair” object, has a location property
  - FoldingChair adds **fold( )** function
  - SwivelChair adds **rotate( )** function

GTK object hierarchy

- GtkButton: pushbutton widget, subclass of...
- GtkBin: widget that contains only one widget, subclass of...
- GtkContainer: container widget for other widgets, subclass of...
- GtkWidget: base object for all widgets, subclass of...
- GObject: base for all objects

How to learn more

- This is where **devhelp** comes in – provides hierarchical listing of inheritance
- For example, let's look at GtkButton in the listing
- Note that GtkButton inherits the properties and methods of the classes above (e.g. “visible“ property from GtkWidget)

Just waiting for you

GTK signals

- This is the basis for interactivity and response
- Main loop and input interaction
- Interaction generates a signal, which can be caught and used to trigger a function
- Example: a button click, a checkbox filled or cleared

Important Python points

- To use GTK library bindings, load using GIR:

```
from gi.repository import Gtk
```

- Create a GtkBuilder object and use **Gtk.Builder.add_from_file( )** to load objects:

```
def __init__(self):
    self.builder = Gtk.Builder()
    self.builder.add_from_file('prog.ui')
    # Now you can refer to elements using
    # the self.builder object
```

Element references

- The **Gtk.Builder.get_object( )** function returns an object based on its name

```
self.close_button =
    self.builder.get_object('close')
```

Connecting & callbacks

- Almost any widget can use the **connect( )** function to map a signal to a callback
- A callback is where you code the reaction to the user's action

```
self.close_button.connect('clicked',
    self.hide_dialog)
```

What's it all about, Alfie?

Simple example

- **easy-entry.py** – Takes text entry from a dialog and outputs the text on the command line
- Demonstrates very simple interface and signals
  - Window, buttons, text entry
  - Window deletion, “clicked” signal
  - GTK main loop

Reminder: Program flow

- XML file defines the GUI elements
- Python code loads file into Gtk.Builder object
- Python code continues to refine GUI and assign interactivity through additional functions

GTK can change the UI

- Everything need not be written in the UI file!
  - Assign label text and other content
  - Hide or show elements
- Example: **PulseCaster**

Parting thoughts

(Apologies to Burt Bacharach for earlier stuff)

- PyGObject doesn't make things harder – concepts are same as in PyGTK
- Try contributing to something before reinventing a wheel
- If you're doing something new, remember: Coding comes last

fedora™

Questions?

http://pfrields.fedorapeople.org/presentations/

Contact:
pfrields@fedoraproject.org