Can your UI change colors like chameleon?

Vikrant Patil
Whats in?
Whats in?

• The talk isn’t really about changing colors! But it’s about rapid UI development.
What's in?

- The talk isn’t really about changing colors! But it’s about rapid UI development.

- At Strand we have build various products on underlying data analysis and visualization platform, avadis(TM). We usually need different wrappers of UI in different products. Although many of them have same underlying core functionality.
Whats in?

• The talk isn’t really about changing colors! But its about rapid UI development.

• At Strand we have build various products on underlying data analysis and visualization platform, avadis(TM). We usually need different wrappers of UI in different products. Although many of them have same underlying core functionality.

• I would like to share tips and tricks that we use at Strand to do rapid UI development.
Patterns in User Interface
Patterns in User Interface: Menus

Visit New File...  (C-x C-f)
Open File...
Open Directory...  (C-x d)
Insert File...     (C-x i)
Close
Save               (C-x C-s)
Save As...         (C-x C-w)
Revert Buffer
Recover Crashed Session
Print Buffer
Print Region
Postscript Print Buffer
Postscript Print Region
Postscript Print Region (B+W)
Postscript Print Region (G+W)
Split Window       (C-x 2)
Remove Splits         (C-x 1)
New Frame            (C-x S 2)
New Frame on Display..  (C-x S 0)
Delete Frame         (C-x S 0)
Exit Emacs           (C-x C-c)

Emacs Tutorial       (C-h t)
Emacs Tutorial (choose language)... (C-h C-t)
Emacs FAQ            (C-h C-f)
Emacs News           (C-h n)
Emacs Known Problems (C-h C-o)
Send Bug Report...
Emacs Psychotherapist
Debian README
Debian News
Debian ChangeLog
Search Documentation
Describe
Read the Emacs Manual (C-h r)
More Manuals
Find Emacs Packages (C-h p)

External Packages

Getting New Versions (C-h C-d)
Copying Conditions (C-h C-c)
(Non)Warranty (C-h C-w)
About Emacs (C-h C-a)
About GNU (C-h C-p)
Patterns in User Interface: Toolbar/Menubar
Patterns in User Interface: Parameters

Connection name: Auto Strand_wirelessSF
- connect automatically
- System setting
  - Wireless Security
    - SSID: Strand_wirelessSF
    - Mode: infrastructure
  - ESSID:
  - MAC address:
    - MTU: automatic
Patterns in User Interface: Parameters
Patterns in User Interface: Parameters
Patterns in User Interface: Custom
Patterns in User Interface

Find most commonly used components. Some of them are as listed below
Patterns in User Interface

Find most commonly used components. Some of them are as listed below

• Menus/Menubars
Patterns in User Interface

Find most commonly used components. Some of them are as listed below

• Menus/Menubars
• Commandbars
Patterns in User Interface

Find most commonly used components. Some of them are as listed below

• Menus/Menubars
• Commandbars
• Parameters input/output
  ○ string parameters
  ○ numeric parameters
  ○ logical/boolean parameters/checkbox
  ○ Dropdown menus etc.
Patterns in User Interface

Find most commonly used components. Some of them are as listed below

• Menus/Menubars

• Commandbars

• Parameters input/output
  ○ string parameters
  ○ numeric parameters
  ○ logical/boolean parameters/checkbox
  ○ Dropdown menus etc.

• Other custom UI components
Automating the patterns
Automating the patterns
Automating the patterns

• Abstract out common parts as core java classes
  ○ e.g Menu, Menubar, Commandbar
Automating the patterns

- Abstract out common parts as core Java classes
  - e.g. Menu, Menubar, Commandbar
- Let the core classes be driven by \textit{specifications} to create the actual UI item.
Automating the patterns

- Abstract out common parts as core java classes
  - e.g Menu, Menubar, Commandbar
- Let the core classes be driven by specifications to create the actual UI item.

These two things will allow us automated UI creation. To generate any UI with new specifications, We need not create the the UI item using basic swing component. What we need is new specification file or whatever way of specifying specifications.
Automating the patterns: Menu

Let us automate the pattern **Menu**. If we want to create new **Menu** item, what we need is a core java class for **Menu** and a specification file. What specifications do you think we will need?
Automating the patterns: Menu

Let us automate the pattern \texttt{Menu}. If we want to create new \texttt{Menu} item, what we need is a core java class for \texttt{Menu} and a specification file. What specifications do you think we will need?

- Name of menu
Automating the patterns: Menu

Let us automate the pattern **Menu**. If we want to create a new **Menu** item, what we need is a core Java class for **Menu** and a specification file. What specifications do you think we will need?

- Name of menu
- List of menu items
Automating the patterns: Menu

Let us automate the pattern Menu. If we want to create new Menu item, what we need is a core java class for Menu and a specification file. What specifications do you think we will need?

- Name of menu
- List of menu items
- Actions of every menu item
Automating the patterns: Menu

Let us automate the pattern Menu. If we want to create new Menu item, what we need is a core java class for Menu and a specification file. What specifications do you think we will need?

- Name of menu
- List of menu items
- Actions of every menu item

Getting Name and list of menu-item s is easy job. It can be taken from any static file. Taking Action s from specification file is little tricky! Jython being a dynamic language can help here.
Let us try to create a new menu \texttt{Python} with only one menu item in it, \texttt{Hello Pycon}

The \texttt{specification} file for this menu in avadis(TM) would look like ...
Automating the patterns : Parameters

specification for various parameters
Automating the patterns : Parameters

**specification** for various parameters

**String**

- Description
- Default value
Automating the patterns: Parameters

**specification** for various parameters

**String**

- Description
- Default value

**Drop down menu**

- Description
- List of items in drop down
- Default value/selection
Automating the patterns: Parameters

**specification** for various parameters
Automating the patterns : Parameters

specification for various parameters

Checkbox

• Description

• Default value
Automating the patterns : Parameters

**specification** for various parameters

**Checkbox**

- Description
- Default value

**Slider**

- Maximum value of slider
- Minimum value of slider
- Default value
I am reapply not a wizard

But let us do some wizardry using Jython spells and just learnt UI automation.
I am reappy not a wizard

But let us do some wizardry using Jython spells and just learnt UI automation.

• Parameter dialogs using Plugins
I am reappy not a wizard

But let us do some wizardry using Jython spells and just learnt UI automation.

• Parameter dialogs using Plugins
• Pycon profile editing form
  ○ From script editor
I am reappy not a wizard

But let us do some wizardry using Jython spells and just learnt UI automation.

- Parameter dialogs using Plugins
- Pycon profile editing form
  - From script editor
  - Add it to menu
I am reappy not a wizard

But let us do some wizardry using Jython spells and just learnt UI automation.

- Parameter dialogs using Plugins
- Pycon profile editing form
  - From script editor
  - Add it to menu
- More magic using Jython
God and Devil

Good
God and Devil

**Good**

- Rapid UI development.
God and Devil

**Good**

- Rapid UI development.
- You can program with efficiency of java and expressiveness of python.
God and Devil

**Good**

- Rapid UI development.
- You can program with efficiency of Java and expressiveness of Python.
- You can do tons of wizardry runtime.

**Bad**
God and Devil

**Good**

- Rapid UI development.
- You can program with efficiency of java and expressiveness of python.
- You can do tons of wizardry runtime.

**Bad**

- There is a chance that your system gets flooded with orphan code snippets in plugins or specification files.
God and Devil

**Good**

- Rapid UI development.
- You can program with efficiency of java and expressiveness of python.
- You can do tons of wizardry runtime.

**Bad**

- There is a chance that your system gets flooded with orphan code snippets in plugins or specification files.
- If action code goes in plugins, syntax errors show up only in runtime.
Thank you!