Emacsy
Shane Celis
GNU Hackers Meeting
Paris, France
August 24th, 2013
Agenda

• Intended Audience
• What is Emacsy?
• What is Emacs?
• Demos
• What’s next?
• Where to get it
Audience

- Interactive application developers
Audience

• Interactive application developers
• Emacs users
Audience

- Interactive application developers
- Emacs users
  - Why base it off Emacs?
Emacsy: Google Summer of Code

- Officially started this Summer
- Working with Ludovic Courtès from GNU Project
- Received lots of help from #guile on freenode.net
- Special thanks to Andy Wingo and Mark Weaver
What is Emacsy?
What is Emacsy?

• It’s an embeddable Emacs-like library.
What is Emacsy?

- It’s an embeddable Emacs-like library.
- How Emacs-like exactly?
A Joke That Explains

Emacs is a great operating system—lacking only a decent editor.
A Joke That Explains

Emacs is a great operating system—lacking only a decent editor.

<table>
<thead>
<tr>
<th>Emacs Extensions</th>
<th>Mail Reader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emacs</td>
<td></td>
</tr>
</tbody>
</table>
A Joke That Explains

Emacs is a great operating system—lacking only a decent editor.
Emacsy Features
Emacsy Features

• lisp interpreter
Emacsy Features

- lisp interpreter
- keymaps
Emacsy Features

• lisp interpreter
• keymaps
• minibuffer
  • history
  • completion
Emacsy Features

- lisp interpreter
- keymaps
- minibuffer
  - history
  - completion
Emacsy Features

• lisp interpreter
• keymaps
• minibuffer
• history
• completion

• recordable macros
Emacsy Features

- lisp interpreter
- keymaps
- minibuffer
  - history
  - completion
- recordable
  - macros
- comprehensive
  - online help
Emacsy Features

- lisp interpreter
- keymaps
- minibuffer
- history
- completion
- recordable macros
- comprehensive online help
- major and minor modes
Emacsy Anti-goals
Emacsy Anti-goals

- Emacsy is not an Emacs replacement
Emacsy Anti-goals

• Emacsy is not an Emacs replacement
• not a text editor
Emacsy Anti-goals

- Emacsy is not an Emacs replacement
  - not a text editor
  - not elisp compatible
Different Integration Strategy

a) Emacs consumes

Emacs

App

b) Emacsy combines

App

Emacsy
What is Emacs?

David T. Ashley

"psp" (pradham.pushkar@gmail.com) wrote in message
news:169453014.587334@12920077@2008.com.googlegroups.com. . .
> I have a very CPU intensive application that should be specified.
> response time. One of my algorithms requires me to do a memcpy() of N
> rows (one row = 2848 elements) very frequently.
> .
> I was wondering how memcpy() is implemented internally, i.e. does it
> involve the CPU or the bus can handle transfer between different
> locations?
> .
> If the rows are in a regular data arrangement, instead of copying the data,
you can change some control information that describes where a given row is.
> For example, one might have an array of pointers, each pointing to a row
> with 2848 elements.
> .
> Instead of moving the elements, just move the pointers around.
> .
> I suggest algorithmic improvement, if possible.
>
> If you're going to be modifying the data, and you need to retain the
> original contents, you need to make a copy.

Barry Margolin

Arlington, MA

*** PLEASE post questions in newsgroups, not directly to me ***
*** PLEASE don't copy me on replies. I'll read them in the group ***
What is Emacs?
What is Emacs?

• It’s a text editor,
What is Emacs?

- It’s a text editor,
- a mail reader,
What is Emacs?

• It’s a text editor,
• a mail reader,
• a shell,
What is Emacs?

• It’s a text editor,
• a mail reader,
• a shell,
• a lisp interpreter,
What is Emacs?

• It’s a text editor,
• a mail reader,
• a shell,
• a lisp interpreter,
• a web browser.
What is Emacs?

Beautiful Architecture. Ch. 11. “GNU Emacs: Creeping Featurism is a Strength” by Jim Blandy
Model View Controller
Model View Controller

- The buffer is a model.
Model View Controller

- The buffer is a model.
- The window is a view.
Model View Controller

• The buffer is a model.
• The window is a view.
• The controller is all the elisp code.
Praising the Model

• A buffer is plaintext.

• Emacs is not a word processor; no bold, italics, nor attributes are embedded into the model.

• It’s simple.

• It’s all just text.
Other Simple Models

- It's all just blocks.
Simple Model

- Users can make very complicated things out of simple building blocks.
Emacs is Extensible

- It’s a text editor,
- a mail reader,
- a shell,
- a lisp interpreter,
- a web browser.
The Fix is In

- The view is fixed.
- The model is fixed.
- Only the controller is extensible with elisp.
Emacsy
Emacsy

• The view is not fixed.
Emacsy

- The view is not fixed.
- The model is not fixed.
Emacsy

• The view is not fixed.
• The model is not fixed.
• The controller is extensible with GNU Guile Scheme.
Vision

Vision

Vision

Vision


Emacsy
Vision


Emacsy
Vision


Emacsy
Vision


Emacsy
**Emacs(y) Architecture**

<table>
<thead>
<tr>
<th>Emacs Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emacs Core</td>
</tr>
<tr>
<td>Elisp Interpreter</td>
</tr>
<tr>
<td>C Core</td>
</tr>
</tbody>
</table>
Emacs(y) Architecture

Emacs Core

Elisp Interpreter

C Core

Emacsy Core

Lisp Interpreter

C/C++ App

Emacs Extensions

Emacsy Extensions
Emacs(y) Architecture

Emacs Core
Elisp Interpreter
C Core

Emacsy Core
Lisp Interpreter
C/C++ App

Emacsy Extensions
Emacs Extensions

Emacsy
Emacs(y) Architecture

Emacs Core
Elisp Interpreter
C Core

Emacsy Core
Lisp Interpreter
C/C++ App

Your App

Emacsy
Demo!
Levels of Integration

1. No buffer support
2. Multiple buffers
3. Multiple windows
What Guile Provides that Elisp Doesn’t
What Guile Provides that Elisp Doesn’t

- Continuations
What Guile Provides that Elisp Doesn’t

• Continuations
• Modules
What Guile Provides that Elisp Doesn’t

• Continuations
• Modules
• Foreign Function Interface (FFI)
What Guile Provides that Elisp Doesn’t

- Continuations
- Modules
- Foreign Function Interface (FFI)
- Threads
Taking Advantage of Guile

- Continuations can be used to build "system-calls" but in userland
- Blocking in Scheme without blocking in C
- Non-blocking events
• In my Emacs environment, I have 6,029 commands available at all times.
• Many aren’t useful in my current context.
• Can we be more careful in what commands we expose?
Context Sensitive Commands

• Idea: Export commands to a module
• Each buffer could have a “MODULE_PATH” that works like PATH in bash
• M-x only looks up commands in the MODULE_PATH
• C-u M-x looks up all commands
Comprehensive Help

• Emacs has a great help system!
• What will Emacsy do?
• Options:
  1. Do it all in minibuffer.
  2. Use a text buffer like Emacs.
  3. Use a local web server.
Beyond Emacs

• Job control
  • Continuations → Coroutines → Cooperative Multitasking

• Major modes
  • Are major modes worth it?
Available Now

- Alpha quality
- Recommend release v 0.1.1
- Available under the GNU GPLv3

https://github.com/shanecelis/emacsys
Thanks for watching!

contact: shane.celis@gmail.com
twitter: @shanecelis
repo: https://github.com/shanecelis/emacsy
Acknowledgments

• GNU Emacs project (Thank you, RMS!)
• GNU Guile project
• The guys at #guile@irc.freenode.net
• Nykia Hunter for the elephant and narwhal graphics
Questions?