

(((IN/Clojure)))

Nov 26, 2016

# Designing an “Object-Functional” System with Clojure

@AdityaAthalye

help*shift*

# Object-Functional?

- A combination of
  - OO ideas, and
  - FP ideas
- For OO people
  - Me going from FP -> OO
  - Clues and References to model things in Clojure
- For FP people
  - Clojure-specific “System” / “Design” things ... later

# A live example

- Automating User Interfaces
  - Web App + Selenium Webdriver
- Develop intuition for “OO” in FP-land through the eyes of
  - Test framework author
    - Vs
  - Test framework users

# Into the minefield...

**REPL-ing through a Web App, as a User**

The image shows a web application interface on the left and its browser developer tools on the right. The web application is titled "New Shared Smart View" and features a search bar, a list of platforms (iOS, Android, Windows Mobile, Windows, Mac, Web, Email), and filters for App, Language, Status, Assignee, and tags. The browser developer tools on the right show the HTML structure of the application, including a select element for platform filtering and a div element for the platform filter. The text "REPL-ing through a Web App, as a User" is overlaid on the image.

Browser Developer Tools (Elements panel):

```
<label>...</label>
<select class="platform-filter" style="display:none;" tabindex="-1" multiple id="yui_3_17_2_1_1480095227907_633" hidden="hidden">
  <option value="ios">iOS</option>
  <option value="android">Android</option>
  <option value="winmobile">Windows Mobile</option>
  <option value="winpc">Windows</option>
  <option value="mac">Mac</option>
  <option value="web">Web</option>
  <option value="email">Email</option>
</select>
<div class="ms-wrapper platform-filter opened" id="yui_3_17_2_1_1480095227907_630">
  <a class="ms-title-wrapper" href="javascript: void(0);" tabindex="1" id="yui_3_17_2_1_1480095227907_1751">...</a>
  <div class="ms-content" id="yui_3_17_2_1_1480095227907_1733">
    <li class="ms-option js-option showing" data_value="all" id="yui_3_17_2_1_1480095227907_647">
      <span class="js-content">All Platforms</span>
    </li>
    <li class="ms-option js-option showing" data_value="ios">...</li>
    <li class="ms-option js-option showing" data_value="android">...</li>
    <li class="ms-option js-option showing" data_value="winmobile">...</li>
    <li class="ms-option js-option showing" data_value="winpc">...</li>
    <li class="ms-option js-option showing" data_value="mac">...</li>
    <li class="ms-option js-option showing" data_value="web">...</li>
    <li class="ms-option js-option showing" data_value="email">...</li>
  </ul>
</div>
<!-- App Filter -->
<label>...</label>
<select class="app-filter" style="display:none;" tabindex="-1" multiple id="yui_3_17_2_1_1480095227907_667" hidden="hidden">...</select>
<div class="ms-wrapper app-filter" id="yui_3_17_2_1_1480095227907_664">...</div>
```

# “Page Objects” (1/2)

The screenshot displays the TARDIS Helpshift admin interface at the URL `tardis.helpshift.mobi/admin/issues/`. The interface is divided into several sections:

- Left Sidebar (Filters):** A dark sidebar with icons for chat, analytics, and settings. It contains filter sections for Platform (All Platforms), App (All Apps), Language (All Languages), Status (All Statuses), Assignee (Everyone), and a section for tags (Issues with no tags, Has all the tags, Has at least one of these tags, Doesn't have any of these tags). A red box highlights the entire sidebar area.
- Issue List:** A list of issues with columns for checkboxes, titles, and timestamps. Two issues are visible: "Are you seeing this, too?" (New Issue 59m) and "If a tree falls in a forest and no one is arou..." (New Issue 1h). A red box highlights this list.
- Conversation View:** The right side shows a conversation with "Mr Robot" (mrrobot@mrrobot.baz). It includes a message "Are you seeing this, too?" and a reply "Mr Robot replied". A red arrow points to the reply.
- Bottom Bar:** Contains buttons for "Enter your reply", "FAQ", "REPLY & RESOLVE", "REPLY", "Unassigned", "ASSIGN TO ME", "NOTE", and "Show Logs". A red arrow points to the "ASSIGN TO ME" button.

# “Page Objects” (2/2)

The screenshot shows the 'NEW FAQ' form in the Tardis Helpshift admin interface. The form is highlighted with a red border. It includes the following elements:

- Published:** A toggle switch currently set to 'OFF'.
- TRANSLATIONS:** A section with a 'English' toggle switch also set to 'OFF'. A red arrow points to this toggle.
- Question:** A text input field with the placeholder 'Question title'.
- Answer:** A rich text editor with a toolbar containing icons for bold (B), italic (i), underline (U), link (A), list (ul), ordered list (ol), link (A), key, image, and video.
- Search Keywords:** A text input field with the placeholder 'Enter keywords' and a help icon (?).
- Instructions:** A note stating 'Use comma "," to separate multiple strings. Strings are not case sensitive.'
- SAVE:** An orange button at the bottom right of the form.

# Language of the Domain

Platform

All Platforms

☐ iOS

☐ Android

☐ Windows Mobile

☐ Windows

☐ Mac

☐ Web

☐ Email

App

All Apps

- Dropdowns
  - exists? visible?
  - open, close, (open?)
  - select, deselect, get-value

2 Issues

↑ Latest

☐ Are you seeing this, too?

☐ If a tree falls in a forest and no one

NEW FAQ

Published

TRANSLATIONS

English

- Checkboxes, Switches
  - exists? visible?
  - select, deselect, (selected?)
  - get-value

# Cross-cutting concerns

	open	close	open?	select	deselect	...
Dropdown						
Checkbox						
Switch						
Modal						
SearchList						
...						
...						



# “Expression Problem”

	List.add	List.get	List.clear	List.size
ArrayList				
LinkedList	Existing method implementations			
Stack				
Vector				
Your new class here	Your new method implmentations			

- OO: easy to add rows

- *new types* extending a *known interface*

	conj	nth	empty	count	
list					Your new function here
vector	Existing implementations				
map					
set					

- FP: easy to add cols

- *new functions* that operate on *existing types*

# And it gets messier...

- Run-of-mill changes
  - CSS changes (tweaks / fixes)
  - Special Cases happen (and variants of special cases)
  - HTML signature change (refactor design)
- ***Everything*** changes (YUI -> React)
- UI = Mutable Soup (incidental complexity)
- Team's needs differ
  - Maintainers v/s Test writers v/s Product

How to Solve It?

# Modelling a “Page Object”

The image displays a web application interface on the left and its corresponding DOM structure on the right, illustrating the concept of a "Page Object" model.

**Web Application Interface (Left):**

- URL: `tardis.helpshift.mobi/admin/issues/`
- Section: **New Shared Smart View**
- Search: `Search in all new issues`
- Filters:
  - Platform: `All Platforms`
  - App: `All Apps`
  - Language: `All Languages`
  - Status: `All Statuses`
  - Assignee: `Everyone`
- Issue List:
  - 2 Issues (Latest)
  - Are you seeing this, too?
  - If a tree falls in a forest and no one is around...
- Issue Details:
  - `div#yui_3_17_2_1_1480094311696_1380.custom-checkbox.inline-checkbox.no-tags-checkbox`
  - ☒ Issues with no tags
- Tags:
  - Has all these tags: `Select tags`
  - Has at least one of these tags: `Select tags`
  - Doesn't have any of these tags: `Select tags`
- Created after: `Select Date`

**DOM Structure (Right):**

The DOM structure shows the HTML elements corresponding to the interface. A red box highlights the `no-tags-checkbox` element, which is a `custom-checkbox` with the ID `yui_3_17_2_1_1480094311696_1380`. The element is a `checkbox` with the type `checkbox` and the ID `no-tags-checkbox`. The label is `Issues with no tags`.

```
<div class="ms-wrapper lang-filter" id="yui_3_17_2_1_1480094311696_319">
  ...</div>
  <!-- Status Filter -->
  <label>...</label>
  <select class="status-filter" style="display:none;" tabindex="-1"
    multiple id="yui_3_17_2_1_1480094311696_539" hidden="hidden">...</select>
  <div class="ms-wrapper status-filter" id=
    "yui_3_17_2_1_1480094311696_536">...</div>
  <!-- Assignee Filter -->
  <label>...</label>
  <select class="assignee-filter" style="display:none;" tabindex="-1"
    multiple id="yui_3_17_2_1_1480094311696_574" hidden="hidden">...</select>
  <div class="ms-wrapper assignee-filter" id=
    "yui_3_17_2_1_1480094311696_568">...</div>
  <div class="separator-line small">...</div>
  <!-- No Tags Filter -->
  <div class="custom-checkbox inline-checkbox no-tags-checkbox" id=
    "yui_3_17_2_1_1480094311696_1380"> == $0
    <input type="checkbox" id="no-tags-checkbox" tabindex="6">
    <span class="checkbox">...</span>
  </div>
  <label class="no-tags-label" for="no-tags-checkbox" id=
    "yui_3_17_2_1_1480094311696_1375">
    <small id="yui_3_17_2_1_1480094311696_1374">Issues with no
      tags</small>
  </label>
  <!-- Has All Tags Filter -->
  <label>...</label>
  <div class="include-tags-filter" tabindex="-1">...</div>
  <!-- Has One Tags Filter -->
  <label>...</label>
  <div class="at-least-tags-filter" tabindex="-1">...</div>
  <!-- Not Tags Filter -->
  <div class="form-field" id="yui_3_17_2_1_1480094311696_1292">...</div>
  <div class="separator-line small">...</div>
  <div class="form-field">...</div>
  <div class="form-field" id="yui_3_17_2_1_1480094311696_1295">...</div>
  <div class="separator-line small">...</div>
  <!-- Smart View Name -->
  <label class="label-name">Name</label>
  <input type="text" id="yui_3_17_2_1_1480094311696_1296" value="New Shared View" />
  </div>
```

# Modelling a “Page Object” Behaviour + Structure

The screenshot shows a web application interface with a dark overlay containing Clojure code. A red arrow points from the title to the code. The code defines protocols for PageObjectify, SelectItems, and GetValue, and a record for Checkbox. The background shows a web application with a sidebar and a main content area.

```
(defprotocol PageObjectify
  (page-object [this])
  (exists? [this])
  (visible? [this])
  (enabled? [this]))

(defprotocol SelectItems
  "~elided~"
  (select [this])
  (selected? [this])
  (deselect [this])
  (reset [this]))

(defprotocol GetValue
  "~elided~"
  (get-value [this]))

(defrecord Checkbox [input-css])

#bugs bunny.util.page_objects.checkboxes.Checkbox
{:input-css "div.smart-view-editor #no-tags-checkbox"}
```

The background shows a web application with a sidebar and a main content area. The sidebar has a search bar and a list of filters. The main content area shows a table of issues.

# What are Clojure Protocols (and Records), and Why?

- Steal *all* of Stu Halloway's *excellent* talk

- [https://gotocon.com/aarhus-2010/presentation/Clojure%20Protocols%20are%20\\*not\\*%20Interfaces!](https://gotocon.com/aarhus-2010/presentation/Clojure%20Protocols%20are%20*not*%20Interfaces!)

- Protocols solve the “expression problem”

- Generic fns
    - arbitrary dispatch
    - + speed + grouping

- Records specify information

- Named type + kwd access + constructors + immutability + RASYDHT
  - e.g. Page Object structure

# Down the REPL rabbit hole...

The image shows a web application interface on the left and its browser developer tools on the right. The web application is titled "New Shared Smart View" and has a search bar "Search in all new issues". Below the search bar is a list of platforms: All Platforms, iOS, Android, Windows Mobile, Windows, Mac, Web, and Email. A red arrow points from the "All Platforms" option to the developer tools. The developer tools show the "Elements" panel with a red box highlighting a specific element. The element is a `<div>` with class `ms-wrapper platform-filter opened` and id `yui_3_17_2_1_1480095227907_630`. Inside this `<div>` is a `<a>` with class `ms-title-wrapper` and href `javascript: void(0);`. The `<a>` is followed by a `<div>` with class `ms-content-wrapper` and id `yui_3_17_2_1_1480095227907_632`. Inside this `<div>` is a `<ul>` with class `ms-content` and id `yui_3_17_2_1_1480095227907_1733`. The `<ul>` contains several `<li>` elements, each with class `option js-option` and a `data_value` attribute. The `<li>` elements are: `<li class="option js-option ms-all showing selected" data_value="all">`, `<li class="option js-option showing" data_value="ios">`, `<li class="option js-option showing" data_value="android">`, `<li class="option js-option showing" data_value="winmobile">`, `<li class="option js-option showing" data_value="winpc">`, `<li class="option js-option showing" data_value="mac">`, `<li class="option js-option showing" data_value="web">`, and `<li class="option js-option showing" data_value="email">`. The `<li class="option js-option showing" data_value="winmobile">` element is highlighted in blue. The developer tools also show the "Console" panel with a message `span.js-content`.

# “Design”-y “System”-y things

- Safety things
  - Invariants, Sane Defaults, & Enforceable Conventions
- People things
  - Separate concerns, Live REPL help, Debugging (printable)
- De-risking development / workflow
  - Known knowns : Design by hammock (in any lang :-)
  - Known unknowns : Design by discovery (fast iteration)
  - Unknown unknowns : Hedge via judicious use of built-in abstractions (remember RASYDHT?)



# Learning Material (1/2)

- Talks / Tutorials

- Stuart Halloway on Clojure Protocols
  - goto conf 2010 : [http://gotocon.com/dl/jaoo-aarhus-2010/slides/StuartHalloway\\_ClojureProtocolsArenotInterfaces.pdf](http://gotocon.com/dl/jaoo-aarhus-2010/slides/StuartHalloway_ClojureProtocolsArenotInterfaces.pdf)
- Sean Devlin's demo on Clojure Protocols
  - clojure conj 2010 : <https://www.youtube.com/watch?v%3DkQhOlWXXl2I>
- Java.next: “Extension without Inheritance”
  - Pt1: <http://www.ibm.com/developerworks/library/j-jn5/index.html>
  - Pt2: <http://www.ibm.com/developerworks/library/j-jn6/index.html>
- Michał Marczyk on Defrecord Deftype in Clojure(Script)
  - Clojure/West '16: [https://www.youtube.com/watch?v=RYGnLf\\_W3tk](https://www.youtube.com/watch?v=RYGnLf_W3tk)

# Learning Material (2/2)

- Books (I've used)

- Programming Clojure
- Joy of Clojure
- Others: Brave Clojure,

- Codebases

- Clojure itself: *e.g. clojure.core.protocols, clojure.data etc...*
- Clj-webdriver: *<https://github.com/semperos/clj-webdriver>*

- REPL / reflection

- (class <foo>), (supers <foo>), (clojure.reflect/reflect <foo>)

# Ideas to use Protocols / Records (1/2)

- Javaland

- Libs for expression-problem scenarios
  - Middleware / wrapper-ware / adapter-ware
  - Low-risk experiments / trials

- Clojureland

- Stuart Sierra's 'component' for
  - app lifecycle mgmt.,
  - dependency injection in immutable world

# Ideas to use Protocols / Records (2/2)

- Clojureland (contd...)
  - Mocks for test suites e.g.
    - (reify MongoDataStore
      - ;; actually do things in-memory
      - (insert [\_ doc])
      - (update [\_ query doc])
      - (upsert [\_ query doc])
      - (delete [\_ query])
      - (commit [\_])
      - (stop [\_]))
  - Etc...

# So...

## *“Designing an 'Object-Functional' System with Clojure?”*

- Design?
  - 'designare' (verb, Latin)
  - mark out, devise, choose / designate / appoint
- Object-Functional?
  - “A combination of OO *ideas* and FP *ideas*”
  - 'cause they're good together
- System?
  - 'systema' (Latin, and Greek)
  - an arrangement / organised whole / compounded of parts
- Clojure?
  - System design + problem-solving mojo

(((IN/Clojure)))

Nov 26, 2016

$\Lambda$

~ fin ~

@AdityaAthalye

helpshift

Extra...