



# Seaside: your next web framework

---

Randal L. Schwartz, [merlyn@stonehenge.com](mailto:merlyn@stonehenge.com)  
Version 1.27 on 20 Jun 2008

This document is copyright 2008 by Randal L. Schwartz, Stonehenge Consulting Services, Inc.  
This work is licensed under Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>



# Why Seaside

- Smalltalk
- Continuation-based
- Persistence and scaling solutions
- Large (and growing) community



# Smalltalk

- Simple to learn
- Mature debugging (including intra-hit)
- Unit testing (including web delivery)
- Templating isn't a separate language
- Multiple vendor choice
- Good team development tools



# Simple to learn

- Entire syntax can be learned in 30 minutes
- Rich libraries included as source
  - Examples of coding
  - Extensible classes (no “primitive” classes)
- Mature means plenty of material to study



# Mature debugging

- Walkback stack is alive
  - Any level gives direct access to source, object, and arguments
  - Edit the source, update, and proceed
- Even intra-hit
  - Walkback in browser triggers debugger
  - Fix the code, proceed
  - And you're still within the same web hit!



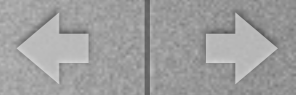
# Unit testing

- Smalltalk community practically invented “Extreme Programming”
- Good testing framework
  - Values are correct
  - Exceptions are thrown properly
- SeasideTesting: full application flow checking



# Templating in Smalltalk

- Paint on an HTML canvas with a brush
  - html heading level: 3; with: 'hello'.
  - html paragraph: 'This text is a paragraph!'.
  - html image url: '/images/foo.jpg'.
  - html image form: aForm.
- Easily generates sane HTML
  - Full access to all standard elements
  - Use Smalltalk tools to refactor, extend



# Multiple vendor choice

- Seaside applications are portable
- Open source
  - Squeak
  - GNU Smalltalk
- Commercial
  - Cincom VisualWorks
  - GemStone/S
  - Instantiations (soon)



# Team development

- Squeak has Monticello
  - Distributed development
  - Good Smalltalk-aware merge tools
  - GemStone/S reads Monticello bundles
- Cincom has Store (but I haven't used it)
- GNU Smalltalk uses files
  - Use your traditional schemes



# Continuations

- Control flow uninterrupted between hits
- Callbacks defined in Smalltalk
- Exceptions can be completely captured



# Control flow

- Inter-hit state is transparently maintained
- You write like the web isn't involved:  
self firstPage.  
[self validate] whileFalse: [self secondPage].  
self thirdPage.
- Execution state is persisted
  - Including all local variables
- Back button can reactivate precise state



# Callbacks

- Anchors show what to do via callbacks:  
html anchor  
    callback: [self exit];  
    with: 'exit'.
- Form elements connect to values:  
html textInput  
    callback: [:e | aValue := e];  
    default: aValue.
- Naming of href and params automatic



# Exceptions

- GemStone/S captures failures
- Logs the error to the Object Log
- Error includes link that activates exception
- Live debugging on the failure
- Works in testing and production



# Persistence

- Traditional object-relational mapper
- Save as objects



# Traditional ORM

- **GLORP mapper (open source)**
  - connects to PostgreSQL and others
  - defaults to “one row is one object”
  - can map many objects or many rows
- **Cincom ActiveRecord (open source?)**
  - builds “normal” GLORP records
  - requires some concessions to schema



# Save as objects

- Squeak ReferenceStreams
  - “write this tree there”
- Magma
  - “watch this tree: save as I change”
  - can cluster to multiple users
- GemStone/S
  - mature large scale persistent VM
  - everything is distributed and fail-over



# Large community

- Active development (Seaside 2.9 in beta)
- Lots of bolt-ons
  - Scriptaculous, YUI, other AJAX
  - Magritte (Metadata), Pier (CMS)
- Active mailing list and website
- Free hosting for simple projects ([seasidehosting.st](http://seasidehosting.st))



# More info

- <http://seaside.st/>
- <http://MethodsAndMessages.vox.com/>