



Seaside

Agile Web Development with Squeak

Overview

I: Introduction: Web Applications

II: Squeak on a Postcard

III: Seaside. A First Overview

IV: Examples, Pointers

Example: Simple Web App

- A very simple "Application"
Add two numbers



What is going on?

first.html

```
<form action="second.html">
  <input type="text" name="value1">
  <input type="submit" value="OK">
</form>
```

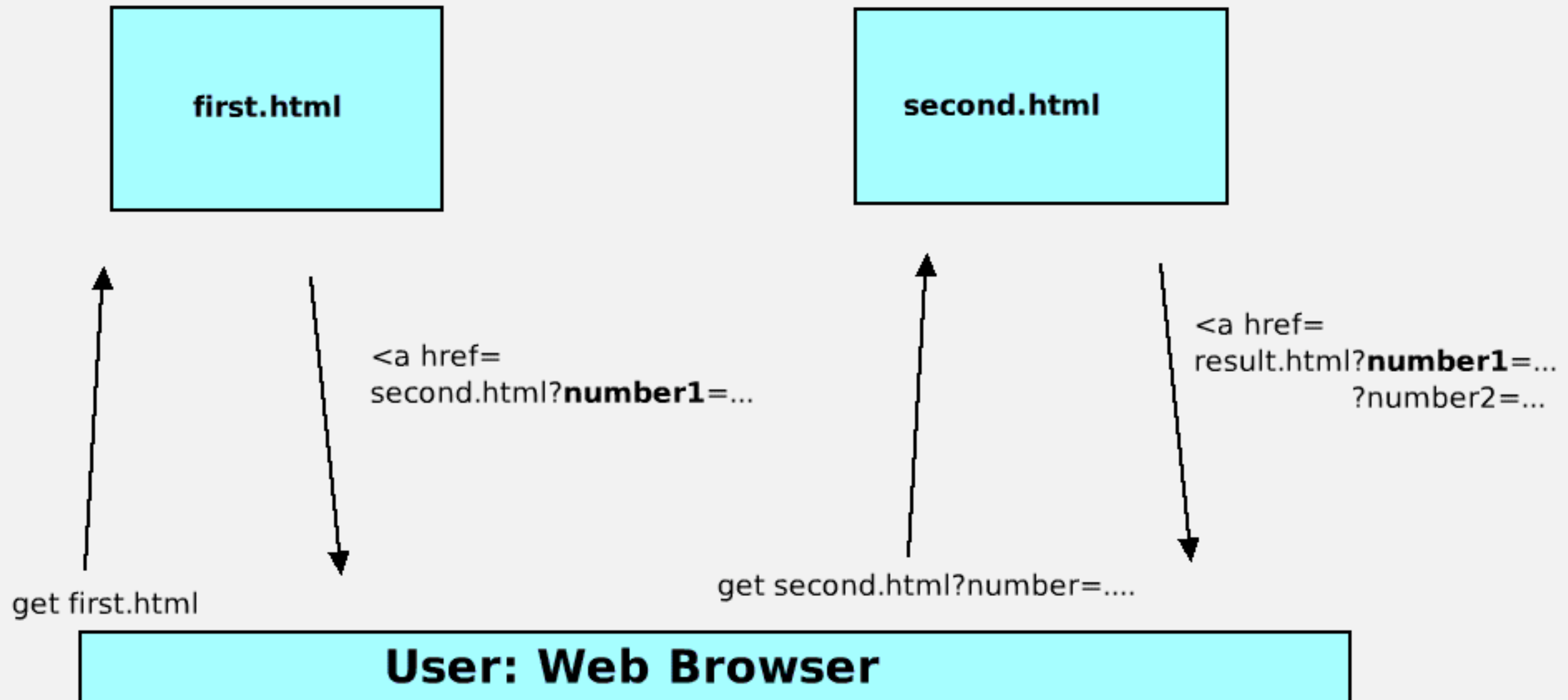
second.html

```
<form action="result.html">
  <input type="hidden" name="value1" value="<% value1 %>" >
  <input type="text" name="value2">
  <input type="submit" value="OK">
</form>
```

result.html

```
<p>
  <% value1 + value2 %>
</p>
```

Control-flow: HTTP request-response



What we want

- Control-flow quite strange.
 - remember GOTO?
- We don't care about HTTP!
- How do you debug that?

And what about

- Back Button?
- Copy of URL?

Example...

- Why not this?

```
go
| number1 number2 |

number1 := self request: 'First Number'.
number2 := self request: 'Second Number'.
self inform: 'The result is ', (number1 + number2) asString.
```

Seaside...

- Session as one piece of code
- HTML / CSS building
- Callback based model
- Composition and reuse
- Debugging and development tools

Part II: Squeak

- Smalltalk overview
- Smalltalk on a postcard
- Examples
- Where is Squeak used?

Squeak: Overview

- 100% Object Oriented:
Everything is an Object
- Dynamically typed
- Virtual Machine, GC
- Huge Class Library
- IDE, Tools for debugging

Smalltalk on a Postcard

exampleWithNumber: x

"A method that illustrates every part of Smalltalk method syntax except primitives."

```
|y|
```

```
true & false not & (nil isNil) ifFalse: [self halt].
```

```
y := self size + super size.
```

```
#$a #a "a" 1 1.0)
```

```
do: [:each | Transcript show: (each class name);  
      show: ' '].
```

```
^ x < y
```

Smalltalk Examples

3 factorial

3 raisedTo: 3

3 + 4

(100 factorial) + 1

(1 < 2) ifTrue: ['yes']

4 timesRepeat: [Beeper beep]

#(1 2 3) select: [:each | each > 1]

Squeak Usage

- Research
- Industry
 - Prototyping
 - Production (mostly with Seaside)
- There is lots of interesting stuff
(Croquet, eToys, 100\$ Laptop,
TK4/Sophie, dabbleDB...)

Part III: Seaside

- Simple example: Counter
- HTML/CSS
- Calls and Callbacks
- Composition and Reuse
- Advanced Debugging and Tools

Example / Tools Overview

- Toolbar
- Halos: Browse/CSS/Inspector

- more on debugging later!



Basics: Task/Component/Rendering

- WAComponent: #renderContentOn:
- WATask: #go
- Components: pages/widgets
 - #call, #answer
- Task: Controlflow of Application

- Framework for generating XHTML
 - Designer: CSS
 - Programmer: HTML
- Have a look at
www.csszengarden.com
- Alternative: Template System
(not actively maintained)

XHTML Building

```
html divNamed: 'title' with: 'Titel'  
html divNamed: 'list; with: [  
    html spanClass: 'item' with: 'Item 1'.  
    html spanClass: 'item' with: 'Item 2'.  
]
```



```
<div id="title">Title</div>  
<div id="list">  
    <span class="item">Item 1</span>  
    <span class="item">Item 2</span>  
</div>
```

Calls: #inform

```
Example3>>renderContentOn: html
  html form: [
    html submitButtonWithAction: [self inform: 'hi']
    text: 'Say hello'.]
```



```
<form method="post" action="/seaside/exampleCallBack1">
<div>
  <input name="_s" value="0hkhwGAdwYxjcfFd" type="hidden" />
  <input name="_k" value="KukciTBW" type="hidden" />
  <input value="Say hello" name="1" type="submit" />
</div> </form>
```

Callback

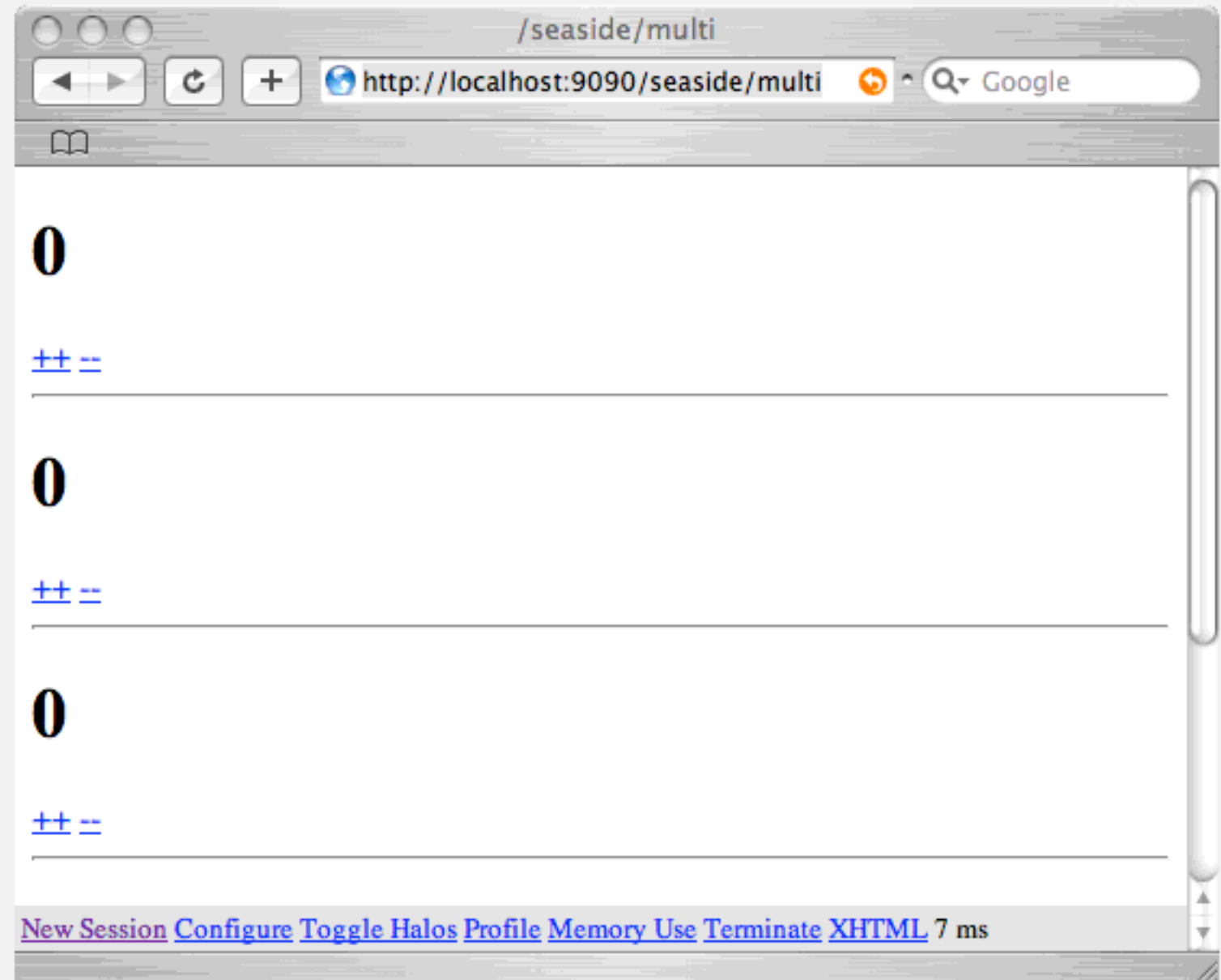
```
renderContentOn: html
  html form: [
    html text: 'Username:'.
    html textInputWithCallback: [:value | username := value].
    html submitButtonWithAction: [self inform: 'Hi', username]
      text: 'Say Hello'.
  ].
```

-Callback:

Code to be executed as a parameter

Composition and Reuse

- Components can be nested



One Piece of Code

- We have seen:
 - Natural control flow
 - Call pages like subroutines
 - return complex objects / callbacks

==> No GOTO-hell!

- Back-button:
 - #isolate: for transactions

Example: Sushi Store

Running Example: Sushi Store

/seaside/store

http://localhost:9090/seaside/store/Anago?_s=zTYnZSc

Google

sushiNet
fresh, raw fish delivered to your door

Search:

[Browse](#)
[Checkout](#)

Anago

Conger Eel

Anago is one of the few pre-cooked sushi. It is grilled and basted with a thick teriyaki sauce. The soft, slighty warm eel, the sweet soy sauce, and the vinegared rice make a particularly delicious sushi.

\$3.00

[Add To Cart](#) [Done](#)

Your cart:
- Anago \$3.00
- Akagai \$3.00
\$6.00

[New Session](#) [Configure](#) [Toggle Halos](#) [Profile](#) [Memory Use](#) [Terminate](#) [XHTML](#) 7 ms

Debugging

- We can inspect, debug and change the running program
- No restart needed!
- Example

There is more...

- Implementation: Continuations...
- Seaside Testing
- Database Access
- Magritte / Pier
- Shore Components
- AJAX... and Scriptaculous support

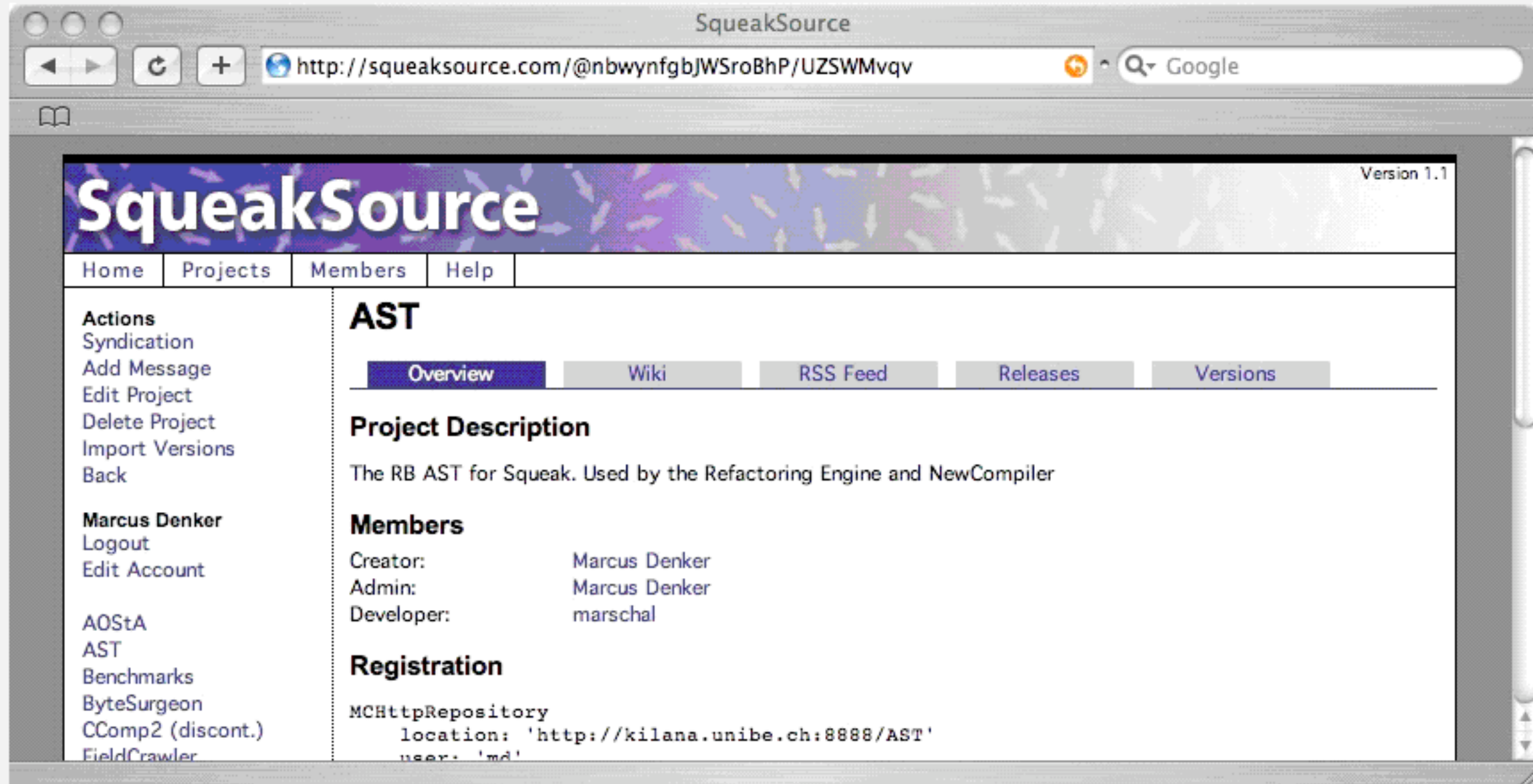
But: No Time!

Part IV: Example Apps / Links

- Some examples
 - SqueakSource
 - Health insurance App
 - DabbleDB

- Links to further information

Example: SqueakSource



The screenshot shows a web browser window titled "SqueakSource" with the URL <http://squeaksource.com/@nbwynfgbJWSroBhP/UZSWMvqv>. The page features a navigation menu with "Home", "Projects", "Members", and "Help". The main content area is titled "AST" and includes a "Project Description" section stating: "The RB AST for Squeak. Used by the Refactoring Engine and NewCompiler". Below this, the "Members" section lists: Creator: Marcus Denker, Admin: Marcus Denker, and Developer: marschal. A "Registration" section shows MCHttpRepository details: location: 'http://kilana.unibe.ch:8888/AST' and user: 'md'. A left sidebar contains a list of actions and links such as "AOSTA", "AST", and "ByteSurgeon".

<http://squeaksource.com>

Example2: Health Insurance

The screenshot shows a web browser window titled "PRO LIFE Offerte" with the URL https://secure.prolife.ch/offerte?_s=PbFhmKfHsrQMWACg&_k=iqEqk. The page has a navigation bar with tabs: "Offertentyp", "Adresse", "Familie", "Zusammenfassung", and "Dokumenten-Druck".

The main content area is divided into two sections:

- Familienmitglieder:** A table with columns "Nachname", "Vorname", "Geburtstag", and "Funktionen". The table is currently empty, with the text "Momentan sind keine Personen erfasst." Below it, there is an information icon and a message: "Hinweis: Sie können Ihre gesamte Familie in einer einzigen Offerte erfassen. Benützen Sie das untenstehende Formular um Personendaten zu erfassen." A button "neue Person hinzufügen" is located below the message.
- Persönliche Daten:** A section with a sub-tab "Versicherungsdeckung". It contains a form titled "Neue Person erfassen" with the following fields:
 - Geschlecht: männlich (dropdown)
 - Name: text input
 - Geburtstag: date input (MM/DD/YYYY)
 - Staatsangehörigkeit: Schweiz (with an "ändern" button)
 - Telefon P: text input
 - eMail: text input
 - Beruf: text input
 - Titel: text input
 - Vorname: text input
 - Zivilstand: dropdown
 - Telefon G: text input
 - Natel: text input
 - Arbeitgeber: text input

<http://offerte.prolife.ch>

Example 3: DabbleDB

The screenshot displays the DabbleDB interface for tracking billable hours. The main view is titled "Work this month, by Employee & Project". It shows a list of entries for two employees: Malcolm Blackwell and Tom Lewis. The entries are categorized by project (Dabble and Hiring) and date. The total billable hours for each employee are shown at the bottom of their respective sections, and a grand total of 36.0 hours is shown at the bottom of the main view.

Employee	Project	Date	Notes	Hours
Malcolm Blackwell	Dabble	October 19, 2005	IE-proofing	9.5
		October 20, 2005	Updated static web pages, importing latest images.	4.5
		October 20, 2005	Random blog-reading.. bound to get something done this way.	3
				17.0
Hiring	October 21, 2005	Checked out downtown Vancouver scene.	8	
				8
				25.0
Tom Lewis	Hiring	October 18, 2005	Read Mythical Man Month; took a surprising number of billable hours!	7
		October 20, 2005	Sat in on this month's Vanlisp meeting.	4
				11
				11
				36.0

<http://www.dabbledb.com>

More information on the web...

- Squeak:

<http://squeak.org>

<http://squeakland.org>

<http://squeak-ev.de>

- Seaside

<http://seaside.st>