Seaside - An Object-Oriented Web Application Development Framework

Seminar
Weiterführende Themen zu Internet- und WWW-Technologien
Sommersemester 2016

Lennard Wolf

Betreuer:
Matthias Bauer, Haojin Yang
Prof. Dr. Christoph Meinel

September 12, 2016

Contents

1 Introduction

Seaside is an open source object-oriented web application development for the Smalltalk programming language. It This text aims at a basic understanding of its history and the context in which it was conceived, its concepts, how they differ from other frameworks of its kind, and what working with it looks like.

This work is structured as follows. Section 2 introduces ... and Section 6 concludes this work.

2 History of Seaside

To understand Seaside's origins, knowledge of the technologies currently popular at the beginnings of its development is required first. This Section will thus start out with such an overview, then present the original and current motivations behind Seaside, followed by a list of the people involved, a development timeline, and finally the current state of both the feature list and the development in general will be examined.

2.1 Historical Context

The viewable internet started out as a network of static pages usually written in HTML. With the progress of time came the want to not only make these sites more visually pleasing, but for the user to be able to interact with them. For this, web application development frameworks have been created for many different programming languages and with differing opinions of what the web of the future should look like. At the beginning of the new millenium, technologies such as *AJAX* (**A**synchronous **J**avaScript **A**nd **X**ML) did not exist, mostly because there had not been a broadly accepted answer to the question, whether web applications should run on the *server* or on the *client* side. Today, in 2016, most web apps run in the users' browsers. In the early 2000s however, many people were sceptical towards the idea of just letting any foreign script run freely on their machine, leading them to block Javascript applications by default. This state of affairs led many developers to build their interactive web applications for server side execution.

During this time, questions like this one, or for example whether URLs have to be human readable, were still up for debate. This is why technologies like Seaside present their own take on them which might appear confusing and non-sensical to some web developers of today.

2.2 Motivation

Over the last few years, some best practices have come to be widely accepted in the web development world. Share as little state as possible. Use clean, carefully chosen, and meaningful URLs. Use templates to separate your model from your presentation.

Seaside is a web application framework for Smalltalk that breaks all of these rules and then some. Think of it as an experiment in tradeoffs: if you reject the conventional wisdoms of web development, what benefits can you get in return? Quite a lot, it turns out, and this "experiment" has gained a large open source following, seen years of production use, and been heralded by some as the future of web applications.

— Avi Bryant, *Web Heresies: The Seaside Framework* [2]

Avi Bryant and Julian Fitzell started working on a web application framework for Smalltalk in 2001 within the context of their consulting endeavours, as well as their current project of building a theatre boxoffice sales system, which was supposed to be using it. Bryant had built such a framework before in Ruby, called *Iowa*, which was heavily influenced by Apple's *WebObjects* for Java. The philosophies shared by these two frameworks were *object-orientation*, *database connectivity*, and *fast prototyping*. These then built the foundations for the new project, which aimed to bring a unique framework to Smalltalk environments.

2.3 Development

The first version, Seaside 0.9, was announced in February 2002 by Bryant and Fitzell to the *squeak-dev* mailing list[1]. It supported action callbacks, a session state system, and a component system, of which the latter two are the core of the Seaside programming principles.

From the beginning it was open source under the MIT License and because of the big amount of feedback from the Smalltalk community, the development could happen quite fast. Thus, version 2.0 came out in October 2003 and was a complete rewrite of the system. It included many new features such as HTML generation, in-browser development tools, and a *Views* layer to name a few. [3]

2.4 Current State

- list of current features:

- Live debugging and code changing while user interacts with web app

- debugger in browser

---

[1]The community is still active today and can be found at `http://lists.squeakfoundation.org/mailman/listinfo/squeak-dev`, accessed: 2016-09-12

- no HTML, render page as Morph -> no browser needed

- CSS, Ajax, Comet, & Javascript support

3 Object-Oriented Web Application Development

In this Section we will shortly introduce the object-oriented programming language Smalltalk in which Seaside is written and used. Then we will discuss the differences between more common approaches to web application development and the object-oriented way.

3.1 Smalltalk

Smalltalk was created in the 70's at Xerox PARC by Alan Kay *et al.*. It is dynamically typed and purely object-oriented, meaning that *everything in a Smalltalk environment is an object* and can thus be easily interacted with. Smalltalk environments such as the free and open source Squeak or the proprietary Gemstone[2] are normally shipped with the Smalltalk core classes (e.g. `Object`) as well as additional ones, depending on the environment's use cases. One of their most important features is *reflection*, with which the programmer is able to inspect and modify every part of the system at runtime. This is especially useful in cases where a system may never shut down, because the changed code does not need to be recompiled but comes into effect immediately.

The syntax can be described as minimalistic. Smalltalk code consists of sequences of messages sent to objects (`anObject message`)[3]. Listing 1 presents syntax basics that are helpful when approaching Seaside.

3.2 Comparison with other technologies

- comparison of features:

- everything is an object

- synchronous vs async

4 Working with Seaside

How does it work[4]

- Session for each user

---

[2]Squeak: `http://squeak.org` Gemstone: `https://gemtalksystems.com`, both accessed: 2016-09-11

[3]A good overview on the Smalltalk syntax was created by Chris Rathman and can be found at `http://www.angelfire.com/tx4/cus/notes/smalltalk.html`

```
"Send the message 'not' to 'true' and assign the returned value to a."
a := true not.

"Messages can be chained."
b := a not not not.

"Blocks are sequences of executable code that return the value of the
last line. They can take any number of arguments (here x and y)."
aBlock := [ :x :y | z := x * y. z - (2 * x) ]

"aBlock is executed with varying parameters depending on b's state."
answer := b ifTrue: [aBlock value: 7 value: 8]
            iFalse: [aBlock value: 0 value: 0].
```

Listing 1: The core syntax of Smalltalk

- ...

- debugging

5 Evaluation

- Each session object can be multiple megabytes big If server crashes, all sessions are lost

- Gemstone distributed but expensive

- Javascript not really debuggable

- Very hard to implement human readable URLs

- Not on par with today's demands of a fast dynamic webpage since its server side

6 Conclusion

References

[1] Daniel G. Bobrow, Kenneth Kahn, Gregor Kiczales, Larry Masinter, Mark Stefik, and Frank Zdybel. Commonloops: Merging lisp and object-oriented programming. In *Conference Proceedings on Object-oriented Programming Systems, Languages and Applications*, OOPLSA '86, pages 17–29, New York, NY, USA, 1986. ACM.

[2] Avi Bryant. "web heresies: The seaside framework" session notes, oscon 2006, July 2006.

[3] Julian Fitzell. Seaside history, 2008.

[4] Michael Perscheid, David Tibbe, Martin Beck, Stefan Berger, Peter Osburg, Jeff Eastman, Michael Haupt, and Robert Hirschfeld. *An Introduction to Seaside - Developing Web Applications with Squeak and Smalltalk*. Hasso-Plattner-Institute, 2008.

[5] Guy L. Steele Jr and Richard P. Gabriel. The evolution of lisp. *ACM SIGPLAN Notices*, 28(3):231–270, 1993.

Figure 1: something something