Software Architecture in an Open Source World

Roy T. Fielding, Ph.D.

Chief Scientist, Day Software
Co-founder, The Apache Software Foundation
Member, W3C Technical Architecture Group
Member, OpenSolaris Community Advisory Board

http://roy.gbiv.com/
Disclaimers

- There is no single “Open Source” model
  - Projects range in scope from the miniscule
    - thousands of code dumps on SourceForge
    - student projects and system dissertations
    - failed commercial ventures
  - to the truly international
    - hundreds of developers
    - collaborating, directly or indirectly
    - on a common platform

- I’ll focus on a subset of “Software Architecture”
  - Run-time architecture, not software structure
  - Realized architecture, not architectural descriptions
  - Principled design for desired properties
## Example open source projects

- World Wide Web
  - URI schemes, HTTP methods, media types
- Linux
  - kernel modules
- Apache httpd
  - feature modules, modular process models, I/O filters
- Mozilla Firefox
  - extensions, themes, XUL, CSS
- Eclipse
  - an architecture of plug-ins
World Wide Web Perspectives

Browsing

Protocols

Information
Web Protocol Extensibility

Uniform Resource Identifiers
- schemes
- hierarchical delegation

HTTP
- versions
- methods
- header fields

Media types
- HTML
- XML
Linux Kernel Modules

Modules

- simplify core
- enable independent development
- promote experiments

Project improves

- reduced friction
- anarchic growth
- more features
- less communication

[diagram from Ivan T. Bowman, 1998]
Apache httpd

- Started with NCSA httpd 1.3
  - Simple, easy to compile on many legacy platforms
  - Limited extensibility via CGI

- Improved security, features, and performance
  - Virtual hosts
  - Pre-forking (adaptive hunt-group) model

- 0.8: re-architected for extensibility (Shambhala)
  - Modular API for features (hook and ladder design)
  - Pools for memory allocation (robustness)

- 2.0: architecture enhanced for more extensibility
  - Modular Process Model (pre-fork, multithreaded, win32, ...)
  - I/O filters and protocol modules
Apache httpd: modules

Modules

- simplify core
- enable independent development
- promote experiments

Project improves

- reduced friction
- anarchic growth
- more features
- less communication

[Apache Modeling Project, f-m-c.org]
Apache httpd: kernel

[Apache Modeling Project, f-m-c.org]
Apache httpd: preforking MPM

[f-m-c.org]
Apache httpd: worker MPM
Apache httpd: winnt MPM
Filters provide more extensibility

- protocol replacement
  - httpd, ftpd, nntpd, ...
- stackable content manipulation
  - extensions that can extend other extensions

[Apache Modeling Project, f-m-c.org]
Mozilla Firefox

Multiplatform

Standards Compliant

Lightweight

Community Supported
Firefox: User-friendly

- Tabbed Browsing
- Integrated Search
- Live Bookmarks
- RSS/XML Feeds
- UI Themes
Firefox: Developer-friendly

Open Source

Extensible Architecture

Plug-in Tools

Layered CSS

Editor Platform
Eclipse Platform

Taking modular extensibility to the next level

Traditional Plug-ins vs. Pure Plug-ins

- Traditional plug-ins
- Pure plug-in system

[Birsan, ACM Queue, Mar 2005]
Eclipse Platform
Eclipse Platform

Unit Tester - Eclipse Platform

Summary:
- Packages: 17% [6]
- Source Classes: 62% [42]
- Test Classes: 62% [42]
- Test Cases: 55% [370]
- Lines covered: 51% [3546]

Execution Summary:

<table>
<thead>
<tr>
<th>Summary of</th>
<th>Successful</th>
<th>Failed</th>
<th>Executed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Classes</td>
<td>62% [42]</td>
<td>38% [26]</td>
<td>100% [60]</td>
<td>60</td>
</tr>
<tr>
<td>Test Classes</td>
<td>62% [42]</td>
<td>38% [26]</td>
<td>100% [68]</td>
<td>68</td>
</tr>
<tr>
<td>Test Cases</td>
<td>55% [370]</td>
<td>45% [300]</td>
<td>100% [673]</td>
<td>673</td>
</tr>
<tr>
<td>Lines covered</td>
<td>51% [3546]</td>
<td>49% [303]</td>
<td>51% [6534]</td>
<td>6534</td>
</tr>
</tbody>
</table>

Executing Test Case: testValueReference2

24 days evaluation.
An open source world

- Most proprietary software projects depend on at least one open source component
  - Internet (bind, httpd, browsers)
  - XML (Xerces, Xalan, Saxon)
  - Scripting (Bash, Perl, Python, Ruby, TCL, Rhino)
  - Security (GPG, OpenSSL, MD5, SHA*)

- And those dependencies are growing
  - Apache Derby (embedded database)
  - Apache Jackrabbit (content repository API)
  - Apache Geronimo (J2EE)
  - Apache Harmony (JVM)
  - Sun OpenSolaris
Why is this important?

- Because innovation doesn’t just “happen”
  - Innovation requires leadership
  - Innovation occurs in spurts
  - Innovation depends on deployment
  - Innovation is aided by extensible architectures

- Because open source is taking the lead
  - Open source encourages collaboration
  - Collaboration is simplified through extensibility
  - Extensibility allows us to stand on the shoulders of giants

- Because it makes Software Research easier!
  - Shared platforms reduce the overhead of systems work
What is common to the largest and most successful open source projects?
- a software architecture
- designed to promote anarchic collaboration
- through extensions
- while preserving control over the core interfaces

Collaborative open source development
- emphasizes community
- takes advantage of the scalability obtainable through Internet-based virtual organizations
- adapts to the volunteer nature of developers

Architecture by design (not a natural byproduct)