Engineering Firefox

Robert O'Callahan
Lizard Engineer
Novell

Novell

About Me

- Graduated Auckland University 1993
- Graduated Carnegie Mellon 2001
- IBM Research mid-2001 to end 2004
  - Research on software development tools
- Worked on Mozilla as hobby, 1999-2004
- Planned return to NZ 2005
  - Looking to work on Mozilla
  - Novell (Ximian) looking for someone

I am a Novell employee but I am not an official representative. All statements and opinions are my own.

All questions welcome.

History

The Netscape Years

- Netscape browser open-source in 1998
  - Impossible to make money with MS giving away IE
- Major code rewrite started late 1998
  - The old code was a real mess ...
- Netscape 6 release late 2000 --- crap!
  - Rightly savaged by press and users; project at nadir
- Mozilla 1.0 release mid 2002
  - Much improved
- Netscape team canned mid 2003, project survives under new Mozilla Foundation

The Mozilla Foundation

- Foundation funded by grants from IBM, AOL, Sun and others, takes over Mozilla development
  - Reduced number of fulltime engineers
  - Makes decisions Netscape/AOL couldn't make, e.g., popup blocking, shift to lean user-friendly Phoenix/Firebird/**Firefox** browser
- Firefox gains market share rapidly
  - Firefox 1.0 release, NYT full-page ad, rave reviews, etc
  - Now >8% in USA, 10-30% in Europe, 25-70% of technical audience
  - Increasing Foundation revenues, many hires
  - Other organizations hiring (e.g. Google, Novell)

Future Challenges

- Microsoft “Avalon” targeted to replace Web
  - IE development suspended
  - Disengaged from W3C
  - Transfer of ideas and staff
  - Microsoft says patent licenses required
- Firefox growing pains
  - Attacker attention
  - Updates etc
  - Higher expectations
- Microsoft reviving IE development
  - Vindication, and threat
- Ongoing struggle to maintain competition and avoid lock-in!

What We're Doing

- Adding new capabilities for Web authors
  - <canvas>
  - SVG
  - CSS columns
  - E4X
  - GPU-accelerated drawing
  - Offline web applications
- New application features
  - Better automatic update
  - Easier management of “stand alone” web apps
  - Aggressive caching for forward/backward
- Performance, bug fixes ad infinitum

Browsers Are Hard

The Web Is Complex

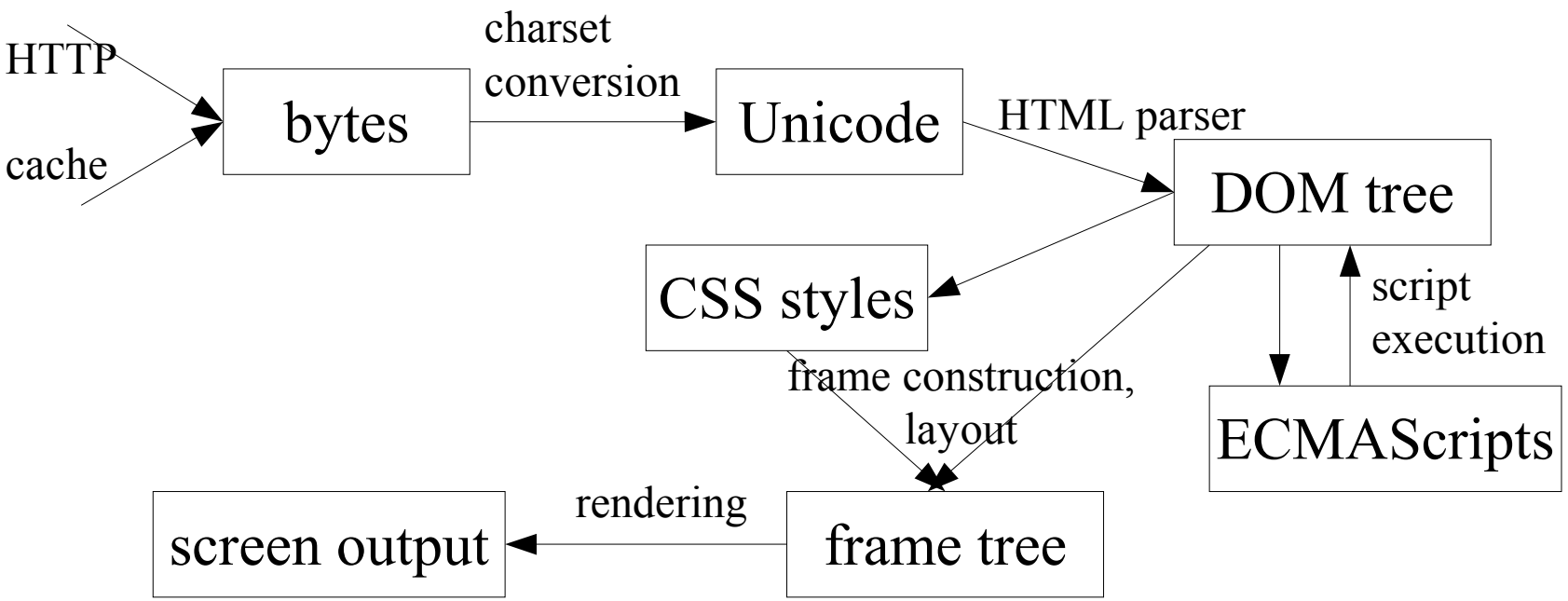
- Standards: HTTP, FTP, SSL, TLS, URL, HTML, XML, XHTML, CSS, JPG, ECMAScript, DOM, GIF, PNG, BMP, ICO, XPath, XPointer, XSLT, RSS, Atom, SVG, XForms, MathML, SOAP, ...
- The “Web as it is”: HTTP', FTP', SSL', TLS', URL', HTML', XML', XHTML', CSS', JPG', ECMAScript', DOM', GIF', PNG', BMP', ICO', XPath', XPointer', XSLT', RSS', Atom', SVG', XForms', MathML', SOAP', ...
- The malicious Web: HTTP , FTP , SSL , TLS , URL , HTML , XML , XHTML , CSS , JPG , ECMAScript , DOM , GIF , PNG , BMP , ICO , XPath , XPointer , XSLT , RSS , Atom , SVG , XForms , MathML , SOAP , ...

Additional Requirements

- Incrementality
- High performance
  - Benchmarks
  - User-perceived
- Ease of use
- Block annoyances (popups, phishing)
- Small download
- Automatic update
- Security: Safety + Liveness
- Satisfy Web authors, server operators, network operators, end users, magazine reviewers, partners, plugin vendors, tool vendors, server vendors, banks

Under The Hood

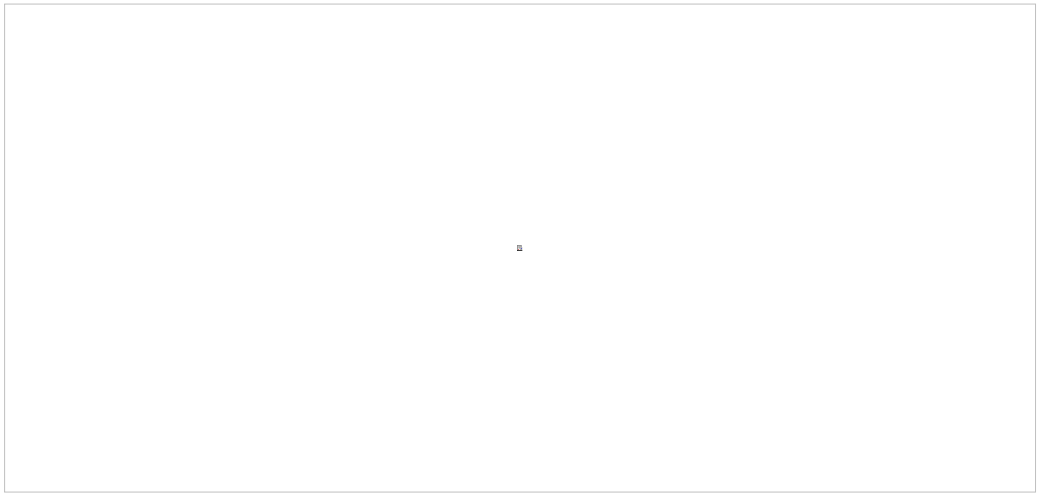
Loading HTML Page



Firefox Architecture

- *Gecko* core implements Web engine
- Firefox UI wrapped around Gecko *content pane*
- Firefox UI is also a Gecko document
  - *XUL* XML dialect
  - Uses JS, DOM, CSS etc
  - Firefox JS talks to C++ components via *XPCOM* IDL interfaces

Firefox By Numbers



and more

Novell.

Beyond Numbers

- Codebase compromised by poor design decisions
  - + Early departure of key designers -> immense difficulties
  - *No-one knows what the invariants are*
  - Gradually being corrected over the years
- Too much reinvention
  - Complex I18N, charset converters, cross-platform graphics, image decoding, HTTP stack
  - 1998: platforms didn't handle these well; today: better
  - Starting to use platform capabilities, deprecate ours
- Constant drive to replace poorly-understood code with well-understood code (preferably someone else's)

Tools And Processes

Source Control

- A few dozen full-time developers
  - Distributed in time and space
- >100 volunteer/part-time contributors
- Good source control essential
- We settle for CVS
- Pros: ubiquitous, reliable, the devil you know
- Cons: too many to list
- High switching cost

Bug Database

- http://bugzilla.mozilla.org
- All changes tracked as bugs (including RFEs)
- Checkin comments link to bugs
- Attach comments, testcases, patches, reviews
- Votes
- Track patch review status
  - Get a list of patches requiring your review
- Track dependencies/regressions
- “Metabugs” track broad issues
- Access control for security issues

More Source Control

- http://bonsai.mozilla.org
- Queryable CVS checkin database
- Online diffs, history
- “CVS Blame”!
- http://lxr.mozilla.org
- Source browsing with lexical markup

Tinderbox

- Supporting many platforms: Win32, OSX, Linux (various), Solaris (various), *BSD, WinCE, AIX, IRIX, QNX, OS/2, BeOS
- And many CPUs: x86, AMD64, ARM, Itanium, MIPS, PPC, Alpha, SPARC, S/390, PA-RISC
- And many compilers: gcc 3.x/4.x, MSVC++ 6.x/7.x, HP, Sun
- Build breakage is a problem!
- http://tinderbox.mozilla.org continuously runs builds on a variety of platforms and monitors status
- Tinderbox provides builds for QA, basic automated regression and performance tests

Code Review

- Code modules have “owner” and “peers”
- All code requires owner or peer review
- “Super-review”
  - Check that code makes sense for project as a whole: cross-module review
  - Initiated to control rogue Netscape module owners
  - Perhaps overly onerous now
- Rules bent when necessary (large crosscutting checkins)
- Difficulties landing large new modules
  - Review delayed until module enabled
  - Review doesn't happen or is weak
- Limited window of visibility into a diff

Testing

- Automated performance tests invaluable
  - 1% performance regression causes alarms
  - Otherwise performance just leaks away over time
- Automated correctness tests less valuable
  - Many regressions not caught
  - Regression tests not complex enough to catch interesting bugs
  - “Obvious” bugs get caught right away...
- Open source community extremely valuable
  - 1000s of downloaders each day
  - Very effective at catching regressions
  - Saving our bacon

Talkback

- Stack traces sent to http://talkback.mozilla.org
- Invaluable for *prioritizing* bugs
- Sometimes helpful for understanding bugs
  - E.g., something common in the environment
  - Need CBI!
- Leaves a blind spot: hangs/freezes

Security

- Shades-of-grey-hats do what they do
  - Fuzz testing
  - Code review
  - Smart poking
- Some get contracted to the Foundation
- Not much structured code auditing
- Modularity helps: string library, charset conversion library, URL parsing APIs, access control check APIs
- Strict adherence to standards helps
  - “Guess what the author means” -> chaos
- JS UI helps

More Security

- Effective automatic update service
- Compiler help (gcc4, MSVC++)
- Want: IE7's “reduced privilege” mode
- Want: Microsoft's “honeymonkey” network
- Automated tools?
  - Coverity: hasn't helped much
  - XPCOM/JS/C++ interactions make life difficult
- Sometimes fixing deep security bugs is harder than finding them
- “Denial Of Service” is a big concern
- Problem: Checking fixes into public CVS reveals bugs early

Language Issues: C++

- Not bad, not great
- Compiler versions vs shifting standard
  - Target a small but growing subset of the language
  - Limit use of templates
  - No exceptions (code bloat!!)
- Optimizer bugs hit with moderate frequency
- Too difficult to create private helper methods
  - Need to add a separate prototype
  - Prototype often needs to go in a public header
  - -> People don't factor code when they should
- Objects grow as you add implemented interfaces
  - Discourages factoring of interfaces

Language Issues: JS

- Not too bad
  - First-class functions, prototype-objects, closures, GC, safe
- Has problems programming-in-the-large
  - No module system
  - Some typechecking or other error detection would be nice
- Adding support for Python scripting
- E4X: First-class XML data in JS

Other Language Issues

- Could we reimplement all of Gecko in a safe modern language?
- Probably not yet
  - Startup performance
  - Need low-overhead real-time GC
  - Tricks to reduce memory footprint
    - E.g. stealing bits from pointers
- It was tried: see Javagator!

Other Tools

- GNU Make
  - Works on all platforms, good enough
- No IDE is really usable
  - Mozilla is too big, too complex, too many languages, custom build process
  - Eclipse CDT is getting close
- Valgrind, Purify helpful
- Profilers helpful
- GDB sucks

Some Open Problems

Multiprocessing

- Multi-CPU machines are the future
  - Really, this time it's true!
- MUST NOT expose threaded programming model
- MUST harness multiple CPUs “under the hood”
- Task parallelism
  - Threads for image decoding, parsing, text measurement
- Parallel rendering?
- Parallel layout???
- *Need ideas here*

Cruft Elimination

```
f(x) { if (x==NULL) return err; ... }
```

- Certain preconditions become true
  - Pointer never NULL
  - Flag never/always set
  - ...
- Weakened precon => bug, fixed by generalizing code
- Strengthened precon => no bug, no change
- Hypothesis: over time, unreachable paths accumulate
- Need powerful source-level dead-code elimination
  - Partial evaluation
- Test hypothesis

Performance Leakage Analysis

- Problem: leak 0.3% of performance every day for a month
- Below statistical noise threshold, hard to pinpoint any individual change
- Very difficult to compare profiles to explain loss
- Configure hardware + software to reduce noise?
- Use simulation to eliminate noise?

Coordinate Reference Frames

`f(nsIFrame* frame, nsPoint pt)`
"`pt` is relative to the origin of `frame`"

- Type systems for units would be helpful, but insufficent
- Types need to say what the origin of the coordinate system is
- Then e.g., for "if ... then pt1 else pt2" require that pt1 and pt2 have the same coordinate system origin
- Frequent bugs in this area
- Not aware of any work in the literature
- Appears to require something fancy

Conclusion

Conclusions

- Hacking Firefox is good ~~clean~~ dirty fun
- Lots of interesting problems at different levels of stack
- Many (most?) problems NOT solved by available languages or tools
- A hard project for tools to digest
- But lots to uncover!
- Save the world!

Novell®

Novell®

Brief History of Mozilla: The State of the Wod

- Rising Firefox market share
  - nearly 10% of average users in USA, > 10% in France, 20% in Germany
  - Organisational rollouts
  - 25-70% of US technical audience
  - Still seeing 200K downloads/day --- straight line
- Increased investment
  - Employing Mozilla developers: IBM, Novell, Sun, Google...
  - Foundation financially strong
  - Key developers who want to be paid, are

Promoting Healthy Competition

- *Browsers still matter!*
  - A core application for most users
  - Or rather, the interface to their applications
  - Safety-critical
    - Vulnerabilities
    - Phishing, scamming etc
  - Usability
    - Interface to a complex and dangerous world
- Without competition, the ruling product stagnates
  - No major revision in IE since 2001
- Rising Firefox share => Microsoft announces IE7

Leveling the Playing Field

- Control over browser can be leveraged to support Web properties (search, portal, etc)
  - E.g., search in IE goes to MSN
- Control over Web standards can favour particular technologies
- Need to open this control point to ensure a level playing field for online businesses

This is for full-screen images

(delete this text)

Novell color palette

These gray lines show the margins that need to be adhered too. If your slide content extends beyond the margins you may need to move to a full screen slide layout.

Most importantly keep the area under the “N” clear

Yellow Palette	Green Palette	Blue Palette	Gray Palette	Misc
R 252 G 194 B 79	R 225 G 213 B 75	R 182 G 201 B 207	R 186 G 189 B 182	R 237 G 238 B 236
R 232 G 128 B 0	R 157 G 176 B 41	R 100 G 132 B 164	R 107 G 108 B 105	R 224 G 0 B 0
R 201 G 86 B 22	R 115 G 126 B 31	R 77 G 68 B 102	R 46 G 52 B 54	R 204 G 0 B 0

Novell

Keeping the Web Open

- IE monopoly ==> risk of everyone just coding to IE
  - Microsoft sets the standards
  - IE required to access the Web
- Strategic issue: non-Microsoft platforms (desktops, phones) permanently locked out of Web
- Issue of principle: no one company should control the standards of communication/information
- Rising Firefox market share => public Web sites cannot afford to be IE-only

Why Mozilla Matters

Mozilla for Desktop Applications

- Mozilla engine can power local applications as well as Web apps
  - Classic Suite, Firefox, Thunderbird all use **XUL**
  - Ongoing work to make this easier for all developers
- Plan: integrate with **Mono**
  - An open source .NET implementation by Ximian/Novell
  - Use .NET languages and tools, plus the convenience and standards-goodness of Web development, all on the portable and widely deployed Mozilla engine
- A rich, efficient, cross-platform, standards-based application platform
- Good for Novell, good for the world

Technical Work

- Moving graphics code to Cairo
  - An open source, modern 2D graphics library
  - Supports hardware acceleration (ab)using 3D/OpenGL
    - David Reveman employed by Ximian
    - Datapoint: Draws text 25-50X faster than GTK2/xlib
- Supporting SVG Web standard for advanced 2D graphics
- Integrating SVG with HTML
  - Add SVG effects to HTML pages
  - E.g., rotating, blended HTML content and forms
- Adding new features to core layout engine
  - Multicolumn text
  - Hyphenation
  - Easier layout

Demo

- <canvas>
- SVG
- http://weblogs.mozillazine.org/roc

Novell.

Novell's Mozilla Activities

- SUSE engineers build and package Mozilla/Firefox
- SUSE and India engineers fixing bugs and improving Firefox integration
  - E.g., make GNOME desktop lockdown preferences to apply to Firefox
- My role
  - Novell's interface to the Mozilla community
  - Make sure Mozilla community schedule aligns with Novell's
  - Get community help for Novell issues
  - Help push Novell's changes back into Mozilla trunk
  - Contribute to Mozilla on Novell's behalf
  - Drive Mozilla forward in directions strategic to Novell

Novell Linux Desktop

- SUSE + Ximian GNOME + trimmings
  - Extra stability work
  - Long support cycle (5 years?)
  - Integration (e.g., themes)
  - Targeted directly at corporate desktop
  - Becoming standard for all users within Novell
- Standard browser: Firefox
  - Best Web compatibility, next to IE
  - Keep familiar applications when moving from Windows
  - Has decent GNOME integration

Novell's Linux Push

Disclaimer: *I know nothing*

- Novell sees open source as a disruptive force, needs new direction
- Jump into open source with two key acquisitions:
  - **SUSE** Linux distribution (primarily servers)
  - **Ximian** Linux desktop
- SUSE is where the money is right now
  - Linux established in servers, not yet on desktops
- Ximian more far-sighted
  - Big money potential in desktops, but harder to penetrate
  - All server businesses vulnerable while Microsoft owns desktop

Mozilla's Future: The Core

- Not enough to just support today's Web apps better
- Need to offer new capabilities to Web developers
  - Graphics-rich applications (like Flash, but extending HTML)
  - Use of hardware graphics acceleration
  - Better text handling (e.g., hyphenation, multi-column)
  - Easier development of rich Web applications
  - Build on existing Web standards and create new ones
- Gmail, Google Maps show what can be done already
  - E.g., use XMLHttpRequest/SOAP to access Web services and eliminate “reload page” problem
- An opportunity to go where Microsoft won't, with better features for Web users and developers

Why Mozilla Matters

The Future

Novell and Mozilla

Round Two: Avalon

- Avalon: Microsoft's new graphics/text/UI framework
  - "XAML" markup language
  - Covers most of HTML, PDF and Flash in one framework
  - Tight integration with Windows, .NET
  - Protected by patents ("license required")
- Will appear for Longhorn and WinXP late 2006
  - "Subset" to be ported to other platforms
- Appears to be designed to supplant Web standards
  - Will establish strong lockin to Windows/.NET, if successful
  - "Indigo" communications framework to tie in servers
- Not healthy

Keeping the Web Open

- IE monopoly ==> risk of everyone just coding to IE
  - Microsoft sets the standards
  - IE required to access the Web
- Strategic issue: non-Microsoft platforms (desktops, phones) permanently locked out of Web
- Issue of principle: no one company should control the standards of communication/information
- Rising Firefox market share => public Web sites cannot afford to be IE-only