

Google Web Toolkit



Technology Analysis

Craftsmanship vs Software Engineering

Craftsmanship

- Small-scale s/w
- Non-critical s/w
- Technology-oriented
- Lower abstraction level

Software Engineering

- Large-scale s/w
- Mission-critical s/w
- Domain-oriented
- Higher abstraction level
- Disciplined, quantifiable

Large-scale s/w

- Large s/w in terms of LOC, classes, etc.
- Many developers, many versions [Parnas]

Secure for mission-critical

- Catch errors as soon as possible (design, compile-time, run-time, on-prod).
- Strong Typing → disciplined, safer s/w
- Compile-time errors → safer s/w
- Unit tests → disciplined, safer s/w
- Metrics, profiling → quantifiable, safer s/w
- Refactoring → multi-versions, maintainability
- IDE aids → disciplined, safer s/w

GWT

Goal: bringing s/w engineering approaches in the world of web development.

Ajax vs GWT (1/3)

- Weak-typing (var)
- Script → errors detected at run-time
- Requires knowledge of browser specific.
- Strong-typing
- Compiled → many errors caught by compl.
- Abstract browser specificities → focus on domain logic
- Language aids: generics, annotations, enums
- IDE aids: refactoring, word completion, syntax coloring, debugging, etc.
- Tools, tools: JUnit, Metrics, Profiler, GUI Builder

Ajax vs GWT (2/3)

- Must choose between small variable names (efficiency) and long meaningful variable names (maintenance)
- Long meaningful variable names in Java source compiled into small variable names → optimization (load time) & obfuscation (security)
- Handles I18N

Ajax vs GWT (3/3)

- Ajax Libraries (jQuery), hard to reuse
- OO Inheritance: facilitate reuse
- Design Patterns (singleton, etc.)
- A lot of libraries: Apache Collections, Log, regex, parser generators, modeling frameworks, etc.
- A lot of IDE plug-ins: Eclipse Find Bugs, Metrics, Web Designer.



Portability w/ Ajax

portable commands

```
if (browser == IE) {  
    //IE-specific  
} else if (browser == FF) {  
    //FF-specific  
} else { ..
```

portable commands

Longer script →
increase load time



Portability w/ GWT

Java source:

portable commands

non-portable commands

portable commands

Browser specificities
hidden behind GWT libs

Gives smaller
generated scripts →
diminish load time

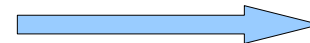


compiled for IE :

portable commands

IE-specific commands

portable commands



compiled for FF :

portable commands

non-portable commands

portable commands



compiled for Chrome



compiled for Safari



compiled for Opera



Efficiency w/ GWT

- Small variables → faster to download
- No IE specifics loaded on a FF browser
- Eliminate dead-code
- Image bundles → (one 100-K image faster to download than 100 x 1K-images)
- Shared validations rules (on client & server) → errors caught by client diminish server load charge



Security w/ GWT

- Browser specificities : tested by Google
- High-level constructs (generics, annotations) → better s/w quality
- Variable obfuscation
- Shared validations rules (on client & server) → a double-check is a safer check
- Unit tests, robustness tests → better quality
- Password hashing on the client → better security



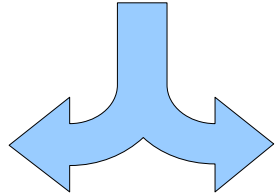
Internationalization w/ GWT

- Properties files
- MessageFormat, DateFormat, etc.
- Locale object
- Google I18N library
- Compilation for a target language

GWT and alternatives

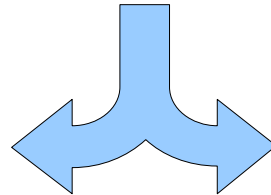
Dynamic web pages

Client: static HTML



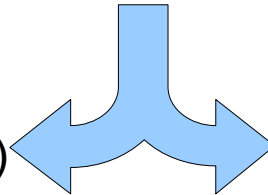
Rich client

Browser plugins: Flash, Silverlight



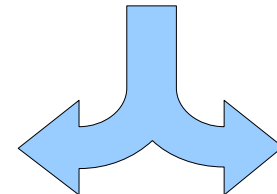
Ajax, HTML5

Ajax scripting (jQuery)



Ajax compilers

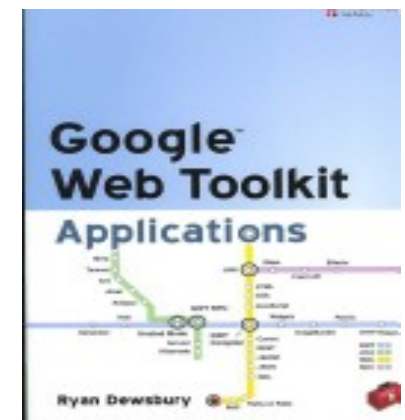
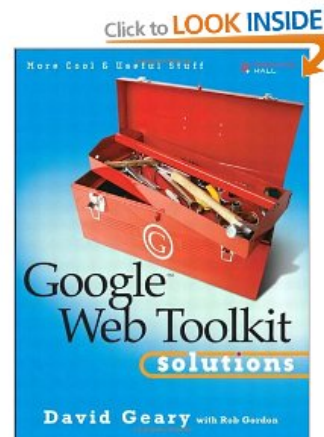
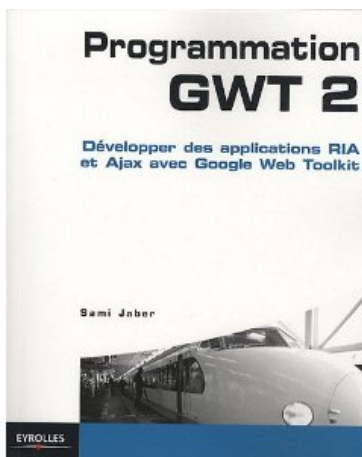
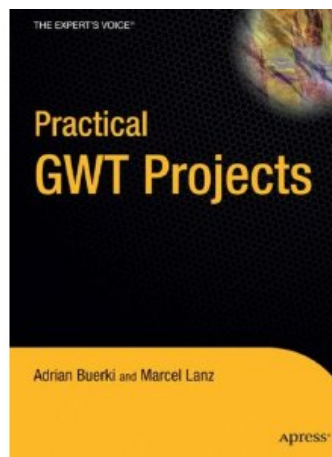
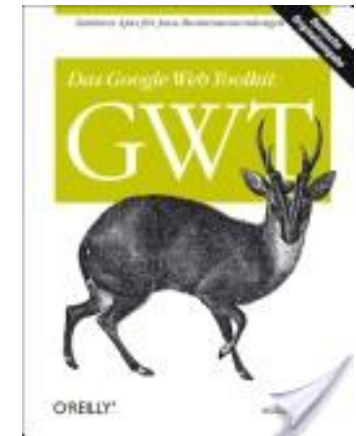
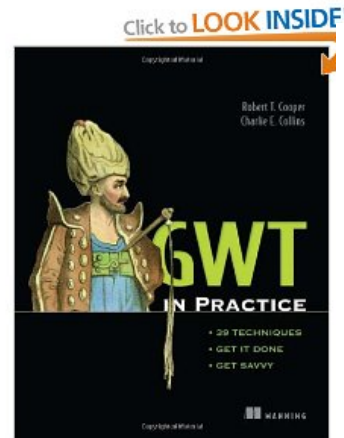
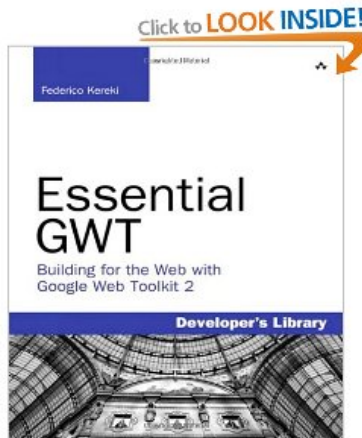
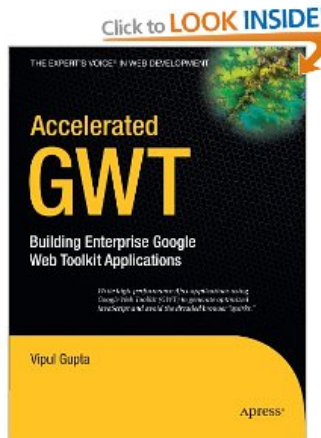
Volta (MS)



GWT not for everything/everyone

- Not appropriate for highly static web pages
- Not appropriate for small-scale projects
- Not appropriate for non-Java and non-Eclipse (or non-NetBeans) developers
- Not appropriate for developers who have not work with a UI library (Swing, SWT) and do not master listeners, layout, etc.

Several GWT Books



References

- Web toolkit: <http://code.google.com/webtoolkit/>
- Showcase: <http://gwt.google.com/samples/Showcase/Showcase.html>
- Roughian: <http://examples.roughian.com/index.htm>

Popularity

- 7th most-popular plug-in on Eclipse Marketplace
- one million downloads the 1st year after its release (ref: [DZone](#)).
- Tens of thousands web sites: Google Moderator, Wave, Go Grid (ref: [Google](#)).
- 164 open source projects tagged GWT on SourceForge

GWT : an active project

- 2010 Oct 28th: version 2.1
- 2009 Dec 8th: version 2.0
- 2009 Sep 22nd: version 1.7
- 2009 Apr 7th: version 1.6
- 2008 Oct 17th: version 1.5
- 2008 Mar 17th: version 1.4
- 2007 Jan 18th: version 1.3
- 2006 May 25th: version 1.0