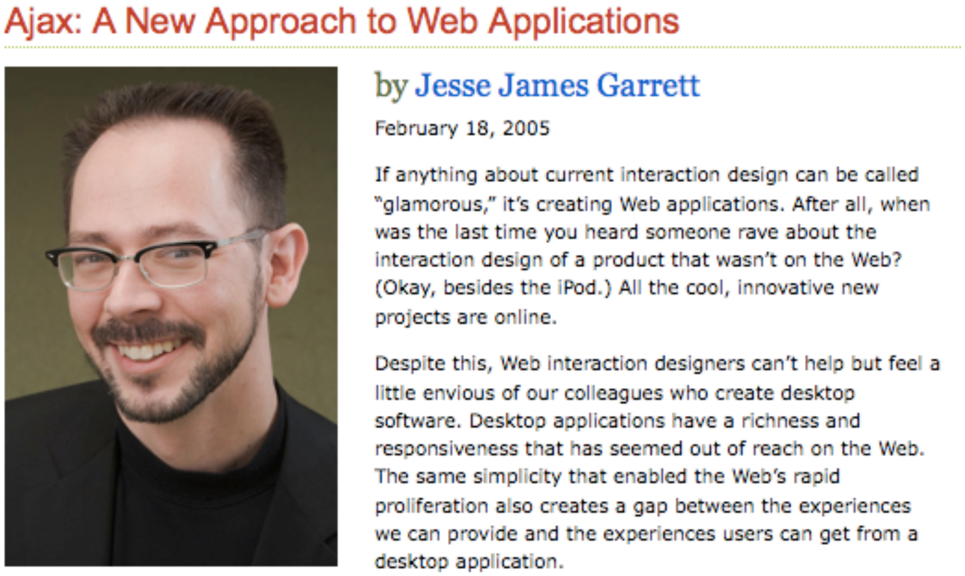
Ajax

Asynchronous JavaScript+CSS +DOM+XMLHttpRequest



Ajax: A New Approach to Web Applications

by Jesse James Garrett

February 18, 2005

If anything about current interaction design can be called "glamorous," it's creating Web applications. After all, when was the last time you heard someone rave about the interaction design of a product that wasn't on the Web? (Okay, besides the iPod.) All the cool, innovative new projects are online.

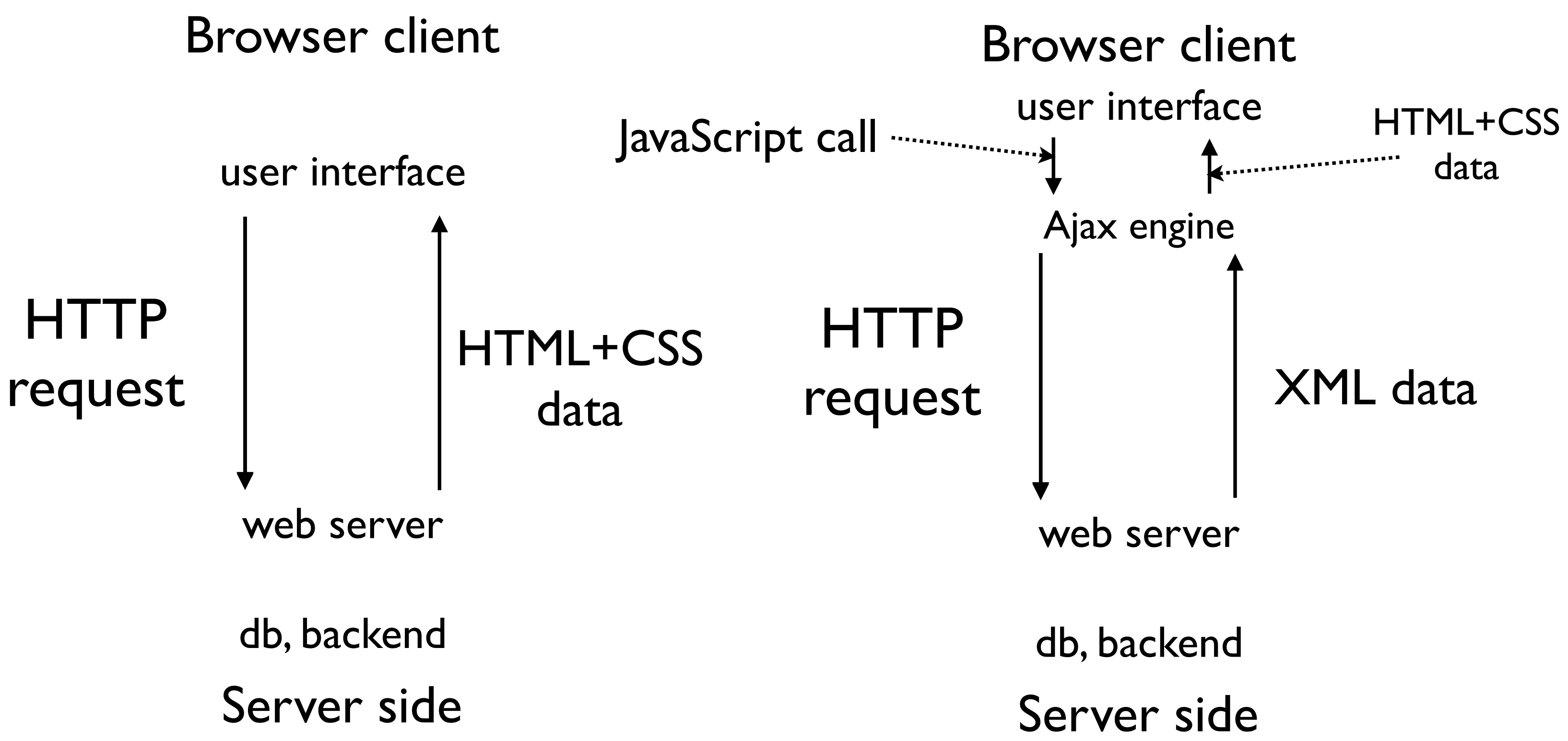
Despite this, Web interaction designers can't help but feel a little envious of our colleagues who create desktop software. Desktop applications have a richness and responsiveness that has seemed out of reach on the Web. The same simplicity that enabled the Web's rapid proliferation also creates a gap between the experiences we can provide and the experiences users can get from a desktop application.

- Coined the term Ajax in 2005
- Describing technology already in use.

What is it?

- Combines several technologies:
  - (X)HTML, CSS - presentation
  - DOM - structured data
  - XML (for interchange)
  - XMLHttpRequest - asynchronous communication
  - JavaScript

Basic idea


Browser client
user interface
HTTP request
HTML+CSS data
web server
db, backend
Server side
Browser client
user interface
JavaScript call
HTML+CSS data
Ajax engine
HTTP request
XML data
web server
db, backend
Server side

History

- Asynchronous loading of content first possible with Java applets (1995)
- XMLHttpRequest - concept created by Microsoft for Outlook WebAccess
- Mozilla implemented it in 2002 (?)
- And there it sat.
- Google suggest - 2004
- And then the idea took off...

General approach

1. Get whatever data you need from the Web form.
2. Build the URL to connect to.
3. Open a connection to the server.
4. Set up a function for the server to run when it's done.
5. Send the request.
6. Response is sent in XML or ...
7. JavaScript - onreadystatechange - takes action

XML or ?

- XML was “the” solution for structured text for a while.
- But it is often more cumbersome than necessary.
- Alternatives?

JSON

- JavaScript Object Notation
- A light-weight data-interchange format
- Subset of JavaScript
  - collection of name value pairs
  - un-ordered list of values
- A JSON object can be eval'd by JavaScript

Example

```
{
   “name” : “Karen Reid”,
   “courses” : 5,
   “address” : {
      “street” : “40 St. George”,
      “city” : “Toronto”
   }
 }
```

When Not to Use Ajax

- If the content never or rarely changes.
- If you don’t see any benefits.
- When it will result in more calls to the server than necessary

When to use Ajax

- When you want to stage in a lot of data or let the user decide which data to bring in.
- Filtering data
- Form validation (but do it on the server too)
- Instant feedback (polls)

Advantages

- More responsive UI
- Potentially fewer connections to the server
  - only changing data needs to be updated

Drawbacks

- Complexity
- Back button
- Bookmarks
- Search engines
- Same origin policy
- Debugging and testing
- Security
- Performance!
- Accessibility