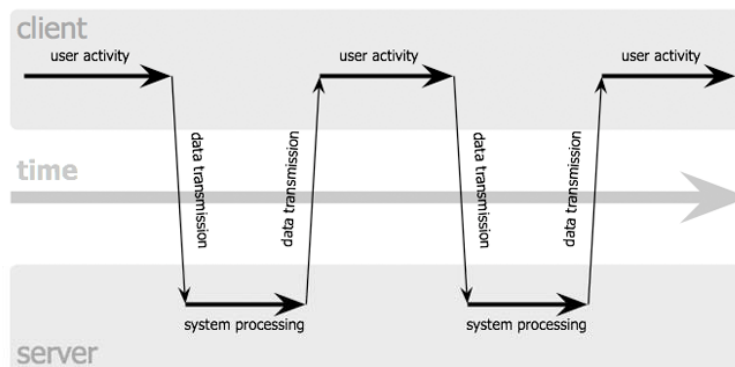


Ajax

- ▣ Techniques for data exchange existed with Java applets in mid-90's
- ▣ 1999 – Microsoft creates the XMLHttpRequest ActiveX control in MSIE5
- ▣ Was introduced into Mozilla and Safari after that
- ▣ Is supported on most platforms and in most browsers
- ▣ Became widely known through Gmail (2003) and Google Maps (2005)
- ▣ Can use languages other than Javascript such as VBScript

Classic Web Exchange

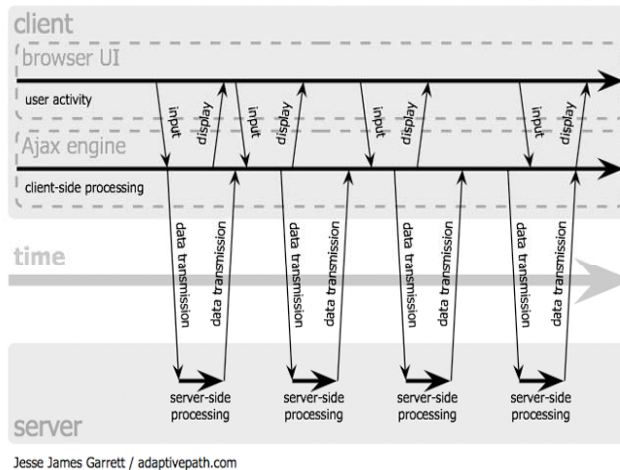
classic web application model (synchronous)



Jesse James Garrett / adaptivepath.com

Ajax Concept

Ajax web application model (asynchronous)



Benefits of Ajax

- Prevents need to reload pages for content refresh
- Introduces new level of interactivity with disparate data sources and dynamic data exchange
- Reduces connections to the server, utilizes caching effectively
- Minimizes bandwidth consumption
- Assists in the maintenance of state rather than the web norm of statelessness

Drawbacks of Ajax

- ▣ Substantially harder to develop
- ▣ Lack of history registration, renders “back” button unusable
 - ▣ Workaround has been developed
- ▣ Makes bookmarking and indexing page content difficult
- ▣ Users who have disabled Javascript or lack XMLHttpRequest will not be able to use it
 - ▣ Phones and small devices suffer from this
- ▣ Suffers from same origin policy
- ▣ Opens browser to attack
- ▣ Can increase number of individual requests on a server
- ▣ High potential for inconsistent web UI

Ajax Code

- ▣ Basic construct of an Ajax exchange

```
//create a variable to hold the object
//default it to false (to indicate NON-existence)
var XHRO = false;
//test if Firefox/Safari method available
//conditional actually says (window.XMLHttpRequest==true)
if (window.XMLHttpRequest) {
    //assign the object to the global variable
    XHRO = new XMLHttpRequest();
//if FF/Safari method not available, test for ActiveX
//instantiation, usually only available in MSIE
} else if (window.ActiveXObject) {
    //assign the variable to the ActiveX
    XHRO = new ActiveXObject("Microsoft.XMLHTTP");
}
```

Ajax Code

- Function to handle data output

```
//add argument to provide the URL of the requested data
function myAjaxCaller(datasource){
  if(XHRO){
    //use the open() method to initiate the request
    //1st argument is HTTP VERB, 2nd is datasource URL
    XHRO.open("GET", datasource);
    //more stuff goes here
    //send the request through the object
    //null is required as an argument in GET requests
    //for POST requests we put form data
    XHRO.send(null);
  }else{
    alert('Sorry, you need an Ajax-capable browser');
  }
}
```

Ajax Code

- Completion handler due to wait-state

```
function myAjaxCaller(datasource){
  if(XHRO){
    XHRO.open("GET", datasource);
    XHRO.onreadystatechange = function() {
      if ( XHRO.readyState == 4 ) {
        //more stuff in here next
      }
    }
    XHRO.send(null);
  }else{
    alert('Sorry, Ajax deficient!');
  }
}
```

- readyState is a property of XHRO
 - 0: uninitialized
 - 1: loading
 - 2: loaded
 - 3: interactive
 - 4: completed

Ajax Code

- ▣ Verification of response

```
if ( XHRO.readyState == 4 ) {  
    var resp = XHRO.responseText;  
    //or var xmlr = XHRO.responseXML;  
    if ( XHRO.status == 200 ) {  
        //ok - do something with the output  
        //maybe send it off to a function  
    } else {  
        //error  
    }  
}
```

- ▣ status is a property of the HTTP response from the server
200: OK
404: missing
304: unchanged
500: server error

Ajax Examples

- ▣ MQ&A Chat Example
- ▣ Google Maps
 - ▣ <http://maps.google.com>
- ▣ Pageflakes
 - ▣ <http://www.pageflakes.com>

Why is Ajax Important?

- ▣ New methodology for data delivery
 - ▣ Uses the advantages of server side processing along with client-side capability
 - ▣ Allows for real-time dynamic data exchange over HTTP
 - ▣ Consumption of RSS and other XML-type resources
 - ▣ Reduction in size individual data exchange providing more room in the pipe for increased quantity of exchanges (more data faster)
- ▣ Requires special attention to semantics
 - ▣ Tagging
 - ▣ POSH works well with Ajax
- ▣ More attention to security and rules
 - ▣ Same origin policy provided new ideas about use of proxies
- ▣ **A foundation of Web 2.0**

What is Web 2.0?

- ▣ A marketing term to describe a new phase of the Web
 - ▣ Originally coined in 1999 by Darci DiNucci in an article “Fragmented Future”¹
 - ▣ Popularized by Tim O’Reilly in 2004
- ▣ Describes an accumulation of changes in technologies used by and uses of the Web itself
 - ▣ Idea being challenged by Berners-Lee – he claims that the Web is just evolutionary by nature, that the fundamental qualitative nature has not changed at all and the term is just hype
- ▣ Not a single shift in the Web itself, described by examples

¹<http://www.cdinucci.com/Darcy2/articles/Print/Printarticle7.html>

Web 2.0 Characteristics

- ▣ “Network as a platform”
 - ▣ Run software-applications entirely through the browser
- ▣ “Architecture of participation”
 - ▣ Users encouraged to add to the content or system (UGC)
 - ▣ Social media
- ▣ Rich Interactive Applications (RIA)
 - ▣ Highly interactive client-side frameworks, including Ajax
- ▣ Syndication
 - ▣ Ability to acquire data from alternate sources for local publication
 - ▣ Signaling data change availability through RSS
- ▣ Tags
 - ▣ Folksonomy – people categorize as they see fit rather than being molded into existing categorical frameworks (as opposed to taxonomy such as Dewey system or Linnaean ranking)

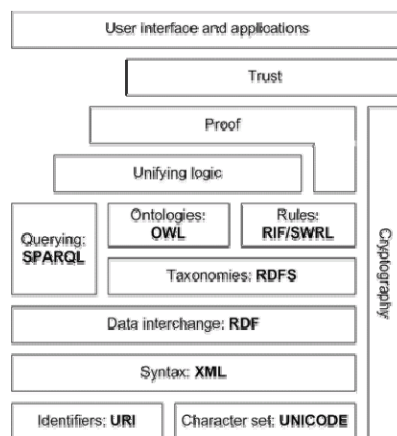
Is Web 2.0 for real?

- ▣ There is enormous criticism about Web 2.0
 - ▣ There is not a change in technology so much as an evolution which coincided with social acceptance and usage
 - ▣ Most of the concepts considered Web 2.0 already existed before
 - ▣ Ajax did not replace underlying HTTP, simply added an abstraction layer
 - ▣ It is, by and large, simply a buzz-word
 - ▣ Removed us from the “dot-com bubble” era
 - ▣ Techno-utopianist idealism – that we’re making the Web to be some mankind-saving technology
 - ▣ The value of the Web is in what it means to users, not the Web itself
 - ▣ Many cultures do not use it or cannot access it (economics)
 - ▣ Social participation yields mistakes

The Next Evolution

- ▣ Semantic Web
 - ▣ Berners-Lee's (endorsed) vision where Web is a universal medium for data, information and knowledge exchange
 - ▣ Principles
 - ▣ Design
 - ▣ Collaborative working groups
 - ▣ Enabling technologies
 - ▣ Intelligent agents (refer to week 1 – a.k.a. Web 4.0)
 - ▣ Content vs. format vs. function – all separate but integral
 - ▣ Resource Description Framework (RDF), Web Ontology Language (OWL) and Extensible Markup Language (XML) allows for arbitrary definition of information as opposed to limitations of HTML – similar to object oriented programming (OOP)
 - ▣ Criticism: “metacrap,” privacy issues, redundancy of data, obsolescence of data, need

The Next Evolution . . .



Semantic Web Stack

- ▣ Challenge
 - ▣ Size
 - ▣ Vagueness
 - ▣ Uncertainty
 - ▣ Inconsistency
 - ▣ Requires widespread adoption of specific flavors of metadata
- ▣ [Metacrap](#)
 - ▣ 2001 - Cory Doctorow

Resources

- ▣ [Wikipedia entry](#) on Web 2.0
 - ▣ Check the references list for specific topics
- ▣ [What is Web 2.0](#) by Tim O'Reilly (oreilly.com, 2005)
- ▣ [Fragmented Future](#) by Darla DiNucci (1999)
- ▣ [Web 2.0 is so over](#) by Jessi Hempel (money.cnn.com, 2009)