



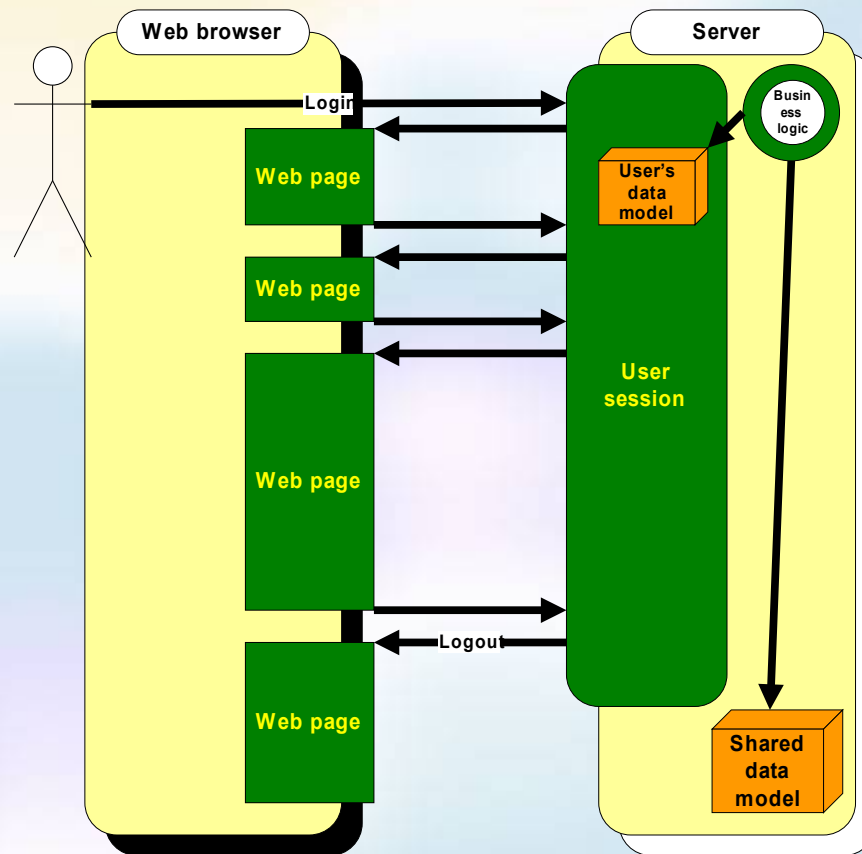
# Using Ajax in ICIS Web Application Development

Alex Cosico & Teri Ulat  
International Rice Research Institute  
Los Banos, Laguna, Philippines

# The State of ICIS Web Application Development

- In the 1<sup>st</sup> generation, we used Perl/CGI with Apache Web server
- Next, we used Tomcat Web server and JSP/JSTL with *Action-oriented* Java-based Web frameworks like *Struts* and *WebWork* for creating dynamic Web apps
- User interaction still somewhat “limited”
- All are “page-based” solutions
- Users experience sometimes long, intermittent delays as the browser moves from one page to the next
- Users want to have more dynamic web apps

# Classic Web App Lifecycle



All the state of the users interaction with the application is held on the web server. The user sees a succession of pages, none of which can do much more except go back to the server to continue the interaction .

# Web 2.0 at a Glance: The Next Generation Web

- “Web 2.0” generally refers to the 2<sup>nd</sup> generation of services available on the World Wide Web
- Aims to give users an experience closer to desktop-based vs. traditional Web-based apps
- Suggests a move away from a mostly static Web or surfing from one website or page to the next, to a more dynamic and interactive Web
- Contributing Web 2.0 technologies include: *weblogs, wikis, podcasts, RSS feeds* and other forms of many-to-many publishing, *social software, web APIs, web standards, web services*, etc.

# What Is Ajax?

- Originally stood for ***Aynchronous JavaScript and XML*** (coined by Jesse James Garret of Adaptive Path in early 2005)
- Not a new technology but a set of *programming techniques* that allow Web apps to communicate with a server in an asynchronous way
- Uses “old” technologies in new ways to create compelling *Rich Internet Applications* (RIA)
- Ajax is a Web 2.0-enabler

# What Does Ajax Bring to the Table?

- Ajax-based Web apps can act on data as soon as the user enters it
- Allows for updating portions of a page instead of always refreshing the entire page
- Result: faster, more responsive, dynamic, and robust Web apps
- Well constructed Ajax apps can rival many desktop-based apps while eliminating existing problems like installation (e.g. clients require only a modern browser) and version control issues (e.g. only the software on the server is updated)

# Some Alternatives

- Adobe/Macromedia Flash
  - Uses a compressed vector graphics format
  - Programmed using ActionScript, a cousin of JavaScript (and both follow the EMAScript standard)
  - Accessed as a browser plug-in; already commonly deployed or plug-in available for most browsers
- Java Web Start
  - A specification for bundling Java-based (heavy) Web apps on a server and which are downloaded and run by the client in a managed “sandbox”
  - Limited distribution at present
- Main downside: both require a runtime component



# Some Principles of Ajax

- The browser hosts an application, not content
  - Classic Web apps: browsers are dumb and stateless
  - Ajax Web apps: hosts and runs an entire JavaScript application
- The server delivers data
  - Classic Web apps: server serves both content and boilerplate
  - Ajax Web apps can send asynchronous requests for new data after initial page load



## Some Principles of Ajax (cont'd)

- User interaction with the application can be fluid and continuous
  - Classic Web apps: browser contacts the server when user clicks on hyperlinks or submits a form
  - Ajax Web apps: browser can contact the server asynchronously in response to a variety of events or user actions
- This is real coding and requires discipline
  - Ajax Web apps must run continuously without breaking or slowing down, possibly with hours of continuous usage, without generating any memory leaks – just like desktop-based client/server apps
  - Can contain complex pieces of code, requiring good structuring of codebases

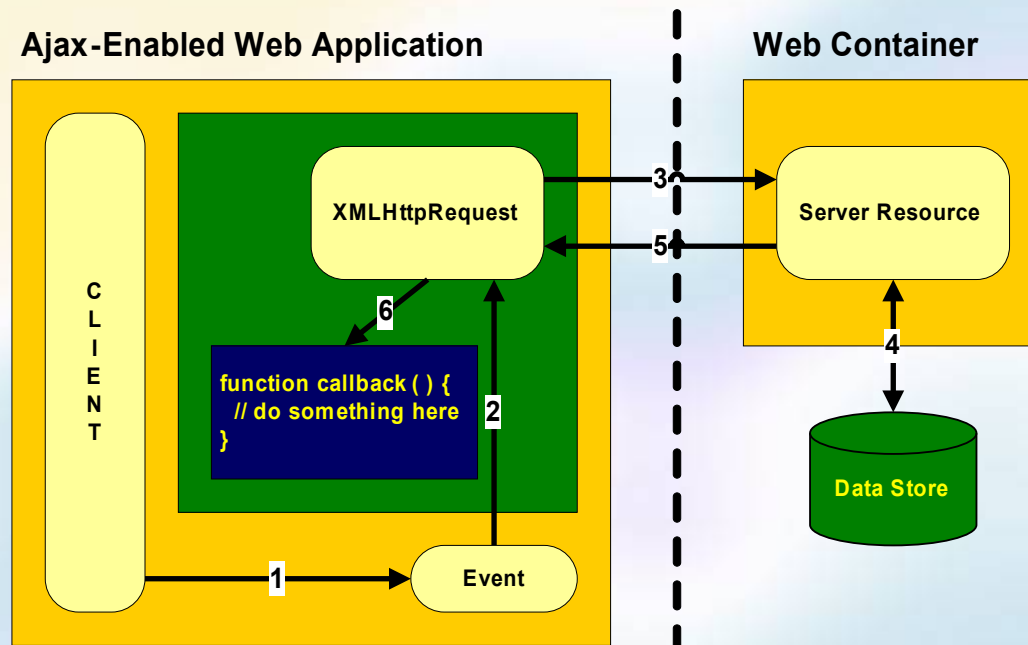
# Key Elements of a Core Ajax App

- An (X)HTML page that contains:
  - UI elements that interact with JavaScript functions
  - JavaScript functions that interact with a server
- Any server-side Web technology that can process HTTP requests and respond in XML markup

# XMLHttpRequest Object

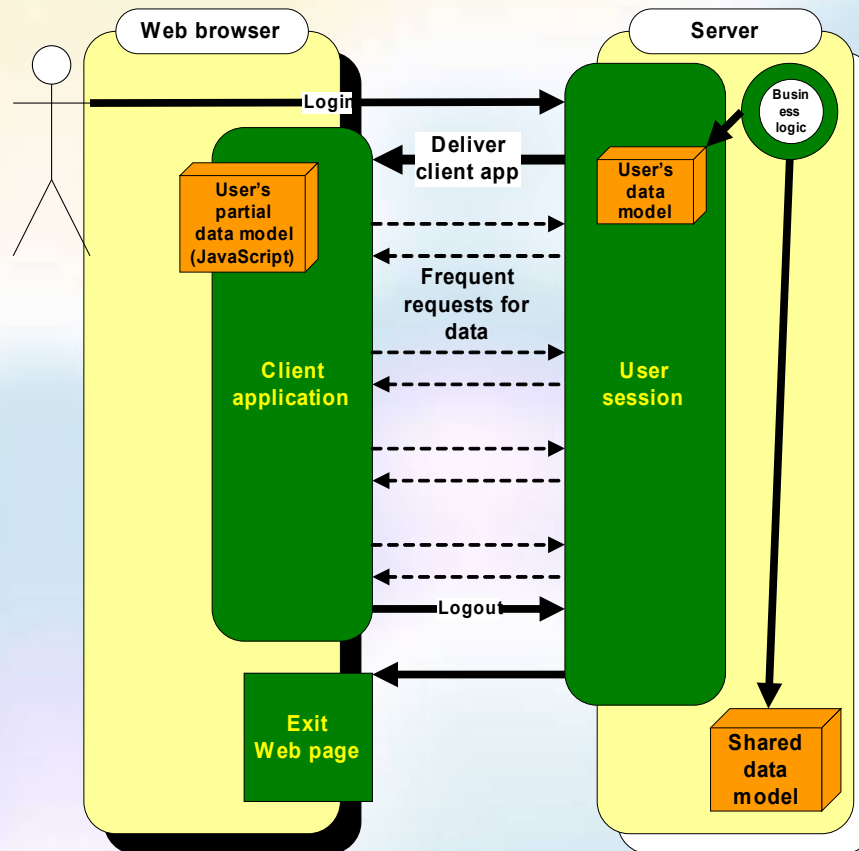
- At the heart of Ajax is the *XMLHttpRequest* object
- Originally implemented by Microsoft in IE 5 (as an *ActiveX* component), but is now supported in most modern browsers
- A *de facto standard*, but not (yet) a W3C standard
- Ajax toolkits like *Dojo* and *Prototype* help simplify cross-browser usage
- Normally used to exchange (but not restricted to) XML data with the server

# A Typical Ajax Interaction



1. A client-side event triggers an Ajax event.
2. An instance of the XMLHttpRequest object is used to open a connection to the server and a callback function is defined that will run after the request completes.
3. A request is made to the server; i.e. a call to a servlet, a CGI script, or any server-side technique.
4. The server does some kind of processing, accessing the database, etc.
5. The request is returned to the browser and the response, often in XML, is parsed and processed by the callback function.

# Ajax Web App Lifecycle



Upon initial login, the client application is delivered to the browser. This application can then field many user interactions, independently and asynchronously, or send requests to the server on the background, without interrupting the user's workflow.

## **Traditional Web App Development vs. Ajax App Development**

- Ajax Web development is much more labor intensive
- Requires an almost complete change in how Web developers approach their work
- In addition to traditional Web design skills, developers require knowledge in:
  - JavaScript and browser event handling
  - DOM
  - CSS
  - XML manipulation

# Ajax Web App Requirements

- Client-side: JavaScript-enabled modern browser (Firefox 1.x, Netscape 8.x, Opera 9.x, Safari 1.2x, MSIE 6.x/7.x)
- Server-side: Virtually any platform that can provide information via HTTP (e.g. Java, PHP, Perl, Ruby, Python, .NET languages, etc.); Ajax is essentially server-side platform agnostic



## Some Examples of Ajax-Based Web Sites and Web Apps

- Google Maps ([maps.google.com](http://maps.google.com)), Google Suggest, Google Mail, Flickr ([www.flickr.com](http://www.flickr.com)), Netflix ([www.netflix.com](http://www.netflix.com)), del.icio.us, digg ([digg.com](http://digg.com)), Technorati ([www.technorati.com](http://www.technorati.com)), Ta-da List ([www.tadalist.com](http://www.tadalist.com)), Chat80 ([www.chat80.com](http://www.chat80.com)), Writely ([www2.writely.com](http://www2.writely.com)), and many more and growing...

## A Case Study: Recommended Lines Web Application

- Client-side: uses Ajax (instead of Struts or WebWork)
- Server-side: uses Java servlets, the Spring Framework, Hibernate
- Backend: IRIS database in MySQL with additional denormalized tables (created using scripts)
- **DEMO**

# A Word on Web Browsers and Web Standards

- Web standards are designed to deliver the greatest benefit to the greatest number of Web users while ensuring the long-term viability of any document published on the Web
- However, most Web browsers currently either do not (yet) implement completely the Web standards (i.e. (X)HTML, CSS, DOM, ECMAScript, etc.), or implement them poorly, or differently

# Caveats in Ajax Web App Development

- Browser support for Web standards are still short of ideal
- Requires extra effort in ensuring Ajax apps that will work with as many Web browsers as possible
- Requires greater effort in testing
- Potential for creation of lots of extra code to address cross-browser compatibility issues
- A relatively young industry for Ajax development toolkits and techniques may require an initially high learning curve and may require lots of experimentation

# Is Ajax for Every Web App?

- **NO!**
- But Ajax functionality can be added to existing Web apps one feature at a time
- Web apps that require a high level of user interaction can benefit from Ajax; otherwise, classic Web app design and tools might be sufficient
- Use Ajax only if it can:
  - Improve application responsiveness and is worth the development effort to implement it
  - Improve the overall end-user experience
- Remember: at the end of the day, and Ajax application is still a Web application

# At a Crossroad in ICIS Web App Development

- Ajax is still a relatively new way of developing ICIS Web apps, but this is the way to go forward
- Ajax design patterns, usability patterns and programming techniques are still evolving
- Software tools and third-party libraries for Ajax development are still at their infancy
- Browsers are still improving with their Web standards compliance
- Despite these, the possibilities and applications for using Ajax in the next generation of ICIS Web apps are many and exciting!

# Some Sites of Interest

- **Ajax News and Info**
  - [ajaxian.com](http://ajaxian.com)
  - [www.ajaxmatters.com](http://www.ajaxmatters.com)
  - [ajaxpatterns.org](http://ajaxpatterns.org)
- **Ajax Tools**
  - [www.dojotoolkit.org](http://www.dojotoolkit.org)
  - [prototype.conio.org](http://prototype.conio.org)
  - [www.turboajax.com](http://www.turboajax.com)
  - [openrico.org](http://openrico.org)
  - [www.clearnova.com](http://www.clearnova.com)
  - [getahead.ltd.uk/dwr](http://getahead.ltd.uk/dwr)
  - [cross-browser.com](http://cross-browser.com)
- **Web Standards**
  - [www.w3c.org](http://www.w3c.org)
  - [www.webstandards.org](http://www.webstandards.org)



## References

- Dave Crane, and Eric Pascarello with Darren James. "***Ajax In Action***". Greenwich, CT: Manning Publications Co., 2005.
- Ryan Asleson, and Nathaniel T. Schutta. "***Foundations of Ajax***". Berkeley, CA: Apress, 2005.
- Nicholas C. Zakas, et al. "***Professional Ajax***". Indianapolis, Indiana: Wiley Publishing, Inc., 2006.



**Q&A**

**Thank You**