ZK
Ajax but no JavaScript

Tom M. Yeh
Founder, the ZK project
http://www.potix.com
> Ajax: Promise and Challenge
> ZK: Simply Rich
> Brand New World
Hot Hot Ajax

> Google Search Results
  > Ajax: 13 million → 93 million in six months
  > Now over 176 millions results

> Ajax >>
User Interface Evolution

[Source: Forrester Resource]
Web the **Only** Platform for Applications

- Ubiquitous Reach
  - Any device, any time and anywhere
  - Globalization and mobility
  - Involvement and collaboration
  - Value = N * Connectivity^2

- Low Total Cost of Ownership
  - Centralized Management
  - No client installation and upgrade
  - Single version
Web applications are great in every measurement, *except* no user wants to use it.
User Frustration

Where are these fields? Where are what I entered?
Challenges of Web Applications

> Inability to visually represent the complexities in today’s applications.
  >> Caused by page-based, submit-and-refresh model
  >> Comparing to desktop apps, Web apps look like terminal apps in 1960s

> Consequence
  >> User frustration
  >> High development costs and risks
Page-based Model of Web Apps

- **Page-based**
  - Submit-and-refresh
  - Stateless

- **Pros**
  - Ubiquitous reach
  - Low TCO

- **Cons**
  - User frustration
  - High development costs and risks

- **Frameworks mostly MVC**
  - Struts, WebWorks, JSF...
Ajax: A New Approach to Web Apps

Ajax = Asynchronous JavaScript + XML

Ajax isn’t a technology. It’s really several technologies, each flourishing in its own right, coming together in powerful new ways. Ajax incorporates:

- **Standards-based presentation** using XHTML and CSS;
- Dynamic display and interaction using the **Document Object Model**.
- Data interchange and manipulation using **XML** and **XSLT**.
- JavaScript binding everything together.
- **Asynchronous data retrieval** using **XMLHttpRequest**.
Ajax: Just One API

```javascript
var req = new XMLHttpRequest();
req.open("POST", "http://killerapp.com/do", true);
req.send("great=true&rich=true");
```
Ajax: Behind-the-Scene Communication

Browser

DOM tree

XMLHttpRequest

Internet

responseXML

Server

Servlet 1

Servlet 2

Servlet 3

Servlet 4
Challenges of Ajax Applications

> AJAX adds more complexities and skill prerequisites to the already costly development of Web applications.

> Incompatible and sophisticated JavaScript API to manipulate DOM and to communicate

> The need to replicate a subset of the application data model and business logic in the browser

> Synchronization between the rich UI at the client and the MVC framework at the server
Synchronization Challenge

When you type the item and click submit, what is the unit price that the server received?

35 or 0.0
You never know!!
Depending on AJAX was responded or not
Programming Skill \textit{Reload}

Windows Programming
Client/Server Programming
Web Programming
J2EE Programming
Ajax Programming
> Ajax: Promise and Challenge
> ZK: Simply Rich
> Brand New World
# Over 130 Ajax Frameworks

<table>
<thead>
<tr>
<th>Feature</th>
<th>ZK</th>
<th>DOJO</th>
<th>Backbase</th>
<th>GWT</th>
<th>Echo2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications running at the client</td>
<td></td>
<td>V</td>
<td>V</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Applications running at the server</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Programming in Java</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programming in JavaScript</td>
<td></td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design in Markup Language</td>
<td>V</td>
<td></td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td>V</td>
<td>V</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Proprietary</td>
<td></td>
<td></td>
<td>V</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Server-side Modal Dialog</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Script in Java</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>V</td>
</tr>
</tbody>
</table>
What Is ZK?

> ZK is an open-source Ajax Web framework that enables rich UI for Web applications with no JavaScript and little programming.

The #1 active project among SourceForge 125,971 projects.

- 1,506 referring sources
- 176 countries
- 18,401 cities
- 119 languages
- 9,838 domains
Why ZK?

> 1993: Beyond Accounting
  > Windows-based
  > Client/Server
  > C++
  > 36 man-months (including a zApp-like framework)

> 2000: RunEC Business Suite
  > Web-based (with some Ajax)
  > N-tier
  > Java
  > 180 man-months (including a Strut+Spring+Hibernate-like framework)
Why a Web App Costs 3~5 Times?

> Intrinsic or the programming model simply wrong?

> Worse of all, no user wants to use it!

> In early 2005, after applying some inventions to a multi-million-dollar project, we believe we got the answer:

**ZK - Simple and Rich**
ZK Architecture

1. JavaScript events
2. ZK requests via HTTP
3. ZK requests
4. ZK requests
5. Update component contents
6. ZK events
7. ZK events
8. EJB, JDO, MQ, Mail…
9. Manipulate components
10. ZK responses
11. ZK responses in XML
12. ZK responses
13. Update via JavaScript
What Make ZK Simple and Rich?

➤ With event-driven and server-centric model, ZK brings back the intuitive programming model to Web developers.

➤ With feature-rich standard-compliant components, ZK enriches Web applications with off-the-shelf building blocks.

➤ With a markup language, ZK makes designing UI as simple as authoring HTML.
Say Hello in ZK

> Creates a file, say, hello.zul under your Web application, e.g., /usr/tomcat/webapps/myapp.

<window border="normal" width="150px">
Hello World!
</window>

> Then, browse to http://localhost/myapp/hello.zul
Say Hello in ZK on Demand

```html
<button label="Say Hello"
        onclick="alert("Hello World!");"/>
```

![Demo of button using ZK to display "Hello World!" dialog box]
ZK Layout Engine

1. Browser sends an URL
2. Load a page
3. Create components
4. ZK events (optional)
5. ZK events (optional)
6. Access persistent layer or other utilities (optional)
7. Manipulate components (optional)
8. Response a HTML page
A Rich Example: Fileupload
Fileupload: UI Design

>window id="imgwnd" title="fileupload demo" border="normal">
   Upload your exciting pictures. <button label="Upload"/>
   <separator/>
</window>
Fileupload: Java Codes

```xml
<window id="imgwnd" title="fileupload demo" border="normal">
  Upload your exciting pictures. <button label="Upload" onclick="upload()"/>
  <separator/>
  <zscript>
    void upload() {
      Object media = Fileupload.get();
      if (media instanceof com.potix.image.Image) {
        Image image = new Image();
        image.setContent(media);
        image.setParent(imgwnd);
      } else if (media != null) {
        MessageBox.show("Not an image: "+media, "Error",
        MessageBox.OK, MessageBox.ERROR);
      }
    }
  </zscript>
</window>
```
Fileupload: MVC

```java
package upload;
public class Uploader extends com.potix.zul.html.Window {
    public void upload() {
        Object media = Fileupload.get();
        if (media instanceof com.potix.image.Image) {
            Image image = new Image();
            image.setContent(media);
            image.setParent(this);
        } else if (media != null) {
            com.potix.zul.html.Messagebox.show("Not an image: "+media,
                                               "Error", Messagebox.OK, Messagebox.ERROR);
        }
    }
}
```
Sample Components

As the Web reaching our daily lives, the effectiveness of Web applications to communicate with users and the simplicity of developing today sophisticated applications becomes more important than ever.

- SIMPLE
- RICH
An UI Designer
contrib by Chris Spiliotopoulos, Athens, Greece
Event Driven Model

> The same event driven model as desktops.
> “Don’t call us. We’ll call you.”

> Like Windows
> Event listeners are invoked sequentially.

> Unlike Windows
> You can suspend and resume event listener anytime.
> You can add and remove event listeners dynamically.
Static Event Listener

```java
public class MyWindow extends com.potix.zul.html.Window {
    public void onOK() {
        //called when user press ENTER
    }
    public void onCancel() {
        //called when user press ESC
    }
}
```
Dynamic Event Listener

class MyListener implements com.potix.zk.ui.event.EventListener {
   public void onEvent(Event event) throws UiException {
      ...//processing the event
   }
   public boolean isAsap() {
      return true; //whether to send back event As Soon As Possible
   }
}

public void init(Window wnd) {
   wnd.addEventListener(new MyListener()); //register an event listener
   ...
Event Threading

- Request Queue (per desktop)
- An event
- Update
- ZK components
- Another request
- Servlet 1
- Event Thread 1
  - Invoke a new thread if Event Thread 1 suspended
- Event Thread 2
- Servlet 2
Use ZK with JDBC

```java
public class MyWindow extends com.potix.zul.html.Window {
    public void onOK() {
        DataSource ds = (DataSource)new InitialContext().lookup("java:comp/env/jdbc/MyDB"); //depend on your configuration
        Connection conn = null;
        Statement stmt = null;
        try {
            conn = ds.getConnection();
            stmt = conn.prepareStatement("INSERT INTO user values(?, ?)" );
            stmt.set(1, name.value);
            stmt.set(2, email.value);
            stmt.executeUpdate();
            stmt.close(); stmt = null;
        } finally { //cleanup
            if (stmt != null) stmt.close();
            if (conn != null) conn.close();
        }
    }
}
```

As a server-centric solution, ZK allows you to use whatever middleware/biz logic you prefer, such as, Hibernate, SAP R/3...
> Ajax: Promise and Challenge
> ZK: Simply Rich
> Brand New World
The Internet Is the Application

- EJB, BPM, MQ, Web Services, SAP R/3, Oracle 10g, Notes
- Any time
  - Any where
    - Any device
- Rich
- Responsive
- Low cost
OSS: Let the World Hear You

> Integrate FCKeditor into ZK: 1 day
> Integrate DOJO into ZK: 1 day
> Integrate jFreeChart into ZK: 2 day
> Visual Designer for ZK: 3 weeks

Leverage Existent Open Source Project, and Put Your Name on It!
Thank you!

ZK
- Ajax but no JavaScript
- The #1 active project in SourceForge.net