



**Rich Internet Applications**

8. Other RIA technologies


---

---

---

---

---

---

---

---

**Ajax applications**

- The Ajax applications we have seen so far have essentially the following characteristics
  - Client tier developed with JavaScript
  - Ajax calls to services which return data as plain text/HTML/JSON/XML
  - Services can be remote, e.g. REST APIs, or local data sources
  - Client code depends on format of service data, not on server technology which produces it
  - Web tier developed with .NET, Java, PHP, etc

Rich Internet Applications 8. Other RIA technologies #2

---

---

---

---

---

---

---

---

**Why use an alternative approach?**

- Need client tier to support functionality not available natively in browser, e.g. rich media
- Want integrated development environment with similar platform/language on client and web tiers
- Want to enhance client data access through closer coordination between client tier JavaScript and web tier objects

Rich Internet Applications 6. Ajax functionality #3

---

---

---

---

---

---

---

---

### Browser plug-ins



- Browser becomes a host for another runtime platform – browser plug-in
- Client code does not rely on browser, so does not need to be JavaScript
- Client code can be developed using whatever languages/frameworks the plug-in supports
- Plug-in renders interface, does not need to be HTML-based

---

---

---

---

---

---

---

---

### Adobe Flex



- <http://www.adobe.com/products/flex/>
- Applications packaged as SWF documents for the Flash Player
- Can also create desktop applications and mobile applications, including for iOS, using Adobe AIR runtime
- Development languages are MXML (for markup) and ActionScript
- Development tool is FlashBuilder

---

---

---

---

---

---

---

---

### JavaFX



- <http://javafx.com/>
- Applications can be packaged as Java applets
- Development language is JavaFX Script which compiles to Java bytecode
- RIAs and desktop applications
- Development tools include the JavaFX SDK and NetBeans

---

---

---

---

---

---

---

---

## Silverlight



- Applications packaged as XAP files for the Silverlight player
- Development languages are XAML (for markup) and .NET languages such as C# and VB.NET
- XAML also used for desktop WPF applications
- Development tools include Visual Studio, Expression Blend
- Note that Adobe (Macromedia) coined the term RIA in 2002

---

---

---

---

---

---

---

---

## Comparison



	Ajax	Flex	Silverlight	JavaFX
Platform	Browser	Flash player	Silverlight	Java
Platform support	Any browser, but there are browser incompatibility issues	Available for most browsers on most platforms, widely installed	Available for most browsers on Windows, Mac, open-source version for Linux	Available for most browsers on most platforms, widely installed
Markup language	XHTML	MXML	XAML	
Scripting language	JavaScript	ActionScript	C#, VB.NET	JavaFX Script
Standard UI components	limited	rich	rich	rich

---

---

---

---

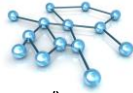
---

---

---

---

## Others



- OpenLaszlo (<http://www.openlaszlo.org/>)
  - Applications can be packaged as SWF files for Flash or as DHTML
- Canoo UltraLightClient (<http://www.canoo.com>)
  - Applications can be packaged as Java applets
- Curl (<http://www.curl.com>)
  - Applications written in Curl language and packaged for Curl RTE platform
- etc...

---

---

---

---

---

---

---

---

### Advantages



- Players can offer richer experience
  - For example, Flash player has built-in support for audio, video streaming, animation, layers, transparency
  - Built-in rich UI components and additional UI toolkits available
- Browser differences don't matter
  - Processing done by plug-in, not by browser
  - Don't need to test with different browser DOMs
- Debugging support in IDE

---

---

---

---

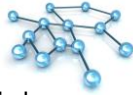
---

---

---

---

### Disadvantages



- Need plug-in to be installed in user's browser
- Potentially large downloads for plug-in and application
- Platform support may be limited
  - e.g. Silverlight not officially available for Linux, although there is an open source implementation (Moonlight), Flash player not available in iOS

---

---

---

---

---

---

---

---

### Silverlight and .NET



- Silverlight contains a subset of the .NET Framework
- Contains components and libraries, including: data integration
  - extensible Windows controls
  - networking
  - base class libraries
  - garbage collection
  - the common language runtime (CLR)

---

---

---

---

---

---

---

---

### Silverlight highlights



- Audio/video
  - Rich media was a key motivation for development of Silverlight
  - Video formats: WMV, H.264 (MP4), VC1
  - Audio formats: WMA, AAC, MP3
- Local storage
  - Uses isolated storage similar to the full .NET framework
  - Filespace within sandbox, accessed through .NET file I/O classes



Rich Internet Applications

6. Ajax functionality #13

---

---

---

---

---

---

---

---

### Silverlight highlights



- Out-of-browser support
  - Can build OOB application which allows elevated privileges compared to in-browser app, including local filesystem access
- Animation
- Drag and drop
  - Users can drag files from My Documents folder to a Silverlight app
- Clipboard access
- Deep Zoom graphics



Rich Internet Applications

6. Ajax functionality #14

---

---

---

---

---

---

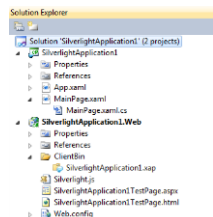
---

---

### Silverlight applications



- Silverlight client project
  - XAML and code-behind files
- Web project which hosts Silverlight app in web page
  - Client app copied as .xap file to ClientBin folder
  - Silverlight.js and test pages contain code to embed app as an object in page



Rich Internet Applications

6. Ajax functionality #15

---

---

---

---

---

---

---

---

### XAML



- Markup to define UI elements and layout
- Databinding can bind control properties to each other or to data

```
<UserControl x:Class="SilverlightApplication1.MainPage"
...
d:DesignHeight="300" d:DesignWidth="400">
<Grid x:Name="LayoutRoot" Background="White">
  <StackPanel Margin="10,10,10,10">
    <TextBox x:Name="txtInput" />
    <TextBlock Text="{Binding ElementName=txtInput,
      Path=Text}" />
  </StackPanel>
</Grid>
</UserControl>
```



---

---

---

---

---

---

---

---

---

---

### Code behind



- Similar to code-behind in Web Forms
- Written in C#, VB
- Event handlers, etc.
- Avoid putting business logic in here
- Should use MVVM pattern to structure application

---

---

---

---

---

---

---

---

---

---

### Building applications



- Implement navigation using Frame and Page controls
- The frame acts as a container for page controls, and facilitates navigation to pages.
- Change the page that is displayed within the frame by navigating, either programmatically or through user action, to a new page.
- Root can contain a combination of navigable content plus permanent user-interface (UI) components

---

---

---

---

---

---

---

---

---

---

### URI mapping



- Within a frame, you can specify that a certain URI pattern maps to a particular page
- URI mapping enables you to create URIs that are descriptive of the user action
- For example, you can specify that any request for /Home is actually a request for the file at /Views/Home.xaml
- Any request that does not match one of the defined patterns is handled as a regular URI request and is not mapped



---

---

---

---

---

---

---

---

### Client-server communication



- Access services from Silverlight code in much the same way as from C# code on server
- Can add service reference in Silverlight project
- Remember that Silverlight code is running on client tier and cross-domain restrictions apply



---

---

---

---

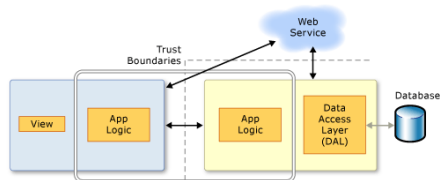
---

---

---

---

### WCF RIA Services



---

---

---

---

---

---

---

---

### WCF RIA services



- Provides closer coordination between client tier and web tier
- Web tier includes Domain Service
- Simple to build a Domain Service class as a wrapper round an EF context, but can use other underlying data sources
- Domain Service exposes IQueryable collections of entities
- Can shape and filter underlying data within service



Rich Internet Applications

6. Ajax functionality #22

---

---

---

---

---

---

---

---

### WCF RIA Services



- Client tier code generated automatically  
Includes client-side context which is a proxy to domain service
- Can query context as if directly accessing underlying context on web tier
- Client context and domain service coordinate communication
- Underlying mechanism is HTTP and binary Xml data format application/msbin1



Rich Internet Applications

6. Ajax functionality #23

---

---

---

---

---

---

---

---

### Silverlight Business Application



- Silverlight Business Application template is installed with WCF RIA Services
- Basic app has navigation and predefined application structure
- In client code behind, add using statement to import web app namespace and build application
- Hidden Generated Code folder created
- See demo code



Rich Internet Applications

6. Ajax functionality #24

---

---

---

---

---

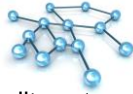
---

---

---



### Using Silverlight



- A. Need client tier to support functionality not available natively in browser, e.g. rich media
- B. Want integrated development environment with similar platform/language on client and web tiers
- C. Want to enhance client data access through closer coordination between client tier JavaScript and web tier objects

---

---

---

---

---

---

---

---

### Should I use Silverlight?



- Good for rich media applications
- Good for complex Line-of-business (LOB) apps
- Many limitations in native browser capabilities addressed to some extent in HTML 5
- Not common in public web applications, users may not have player installed
- Microsoft's continued support and development is uncertain
- Similar story for Flex, JavaFX

---

---

---

---

---

---

---

---

### Google Web Toolkit



- Allows you to write your AJAX front-end in Java
- GWT then cross-compiles into optimized JavaScript to be deployed on web site
- Can develop and debug in the same development environment
- Claims to produce JavaScript that loads and executes faster than equivalent handwritten JavaScript
- Client-server communication via RPC

---

---

---

---

---

---

---

---

### Using GWT



- A. Need client tier to support functionality not available natively in browser, e.g. rich media
- B. Want integrated development environment with similar platform/language on client and web tiers
- C. Want to enhance client data access through closer coordination between client tier JavaScript and web tier objects

---

---

---

---

---

---

---

---

### Web remoting



- Uses client-side libraries which abstract details of Ajax calls and integrate with a specific server-side technology
- Client code written as though it directly calls server side code

---

---

---

---

---

---

---

---

### Web remoting



- Client side code written in JavaScript
- Uses client-side libraries integrate with a specific server-side technology
- Create client-side proxy classes to make calls to server side objects
- Similar to RPC web service call, except not intended for public web services
- Unlike SOAP services, don't use WSDL to advertise methods

---

---

---

---

---

---

---

---

## DWR and WebORB



- Direct Web Remoting (DWR)
  - JavaScript on client
  - Java on server
- WebORB
  - JavaScript, Flash, Flex on client
  - Java, .NET, PHP, Ruby on Rails, web services on server side

---

---

---

---

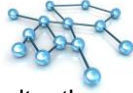
---

---

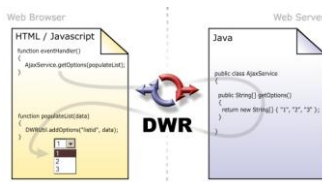
---

---

## DWR



- The diagram shows how DWR can alter the contents of a selection list as a result of some Javascript event




---

---

---

---

---

---

---

---

## DWR example - server



```

Java source:
package org.getahead.dwrdemo.simpлетext;

public class Demo {
    public String sayHello(String name) {
        return "Hello, " + name;
    }
}
    
```

server method is **Demo.sayHello** which returns a string

```

dwr.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dwr PUBLIC
    "-//GetAhead Limited//DTD Direct Web Remoting 2.0//EN"
    "http://getahead.org/dwr/dwr20.dtd">
<dwr>
  <allow>
    <create creator="dwr" javascript="Demo">
      <param name="class" value="org.getahead.dwrdemo.simpлетext.Demo"/>
    </create>
  </allow>
</dwr>
    
```

server configuration file causes JavaScript proxy class to be created

Example from <http://directwebremoting.org/>

---

---

---

---

---

---

---

---

### DWR example - client



**HTML source:**

```
<p>
  Name:
  <input type="text" id="demoName"/>
  <input value="Send" type="button" onclick="update()"/>
  <br/>
  Reply: <span id="demoReply"></span>
</p>
```

**Javascript source:**

```
function update() {
  var name = dwr.util.getValue("demoName");
  Demo.sayHello(name, function(data) {
    dwr.util.setValue("demoReply", data);
  });
}
```

callback function declared inline in this example

call server method – actually calls method of JavaScript proxy class **Demo.js**, proxy and DWR engine handle transport of data in both directions

---

---

---

---

---

---

---

---

---

---

### Using web remoting



- A. Need client tier to support functionality not available natively in browser, e.g. rich media
- B. Want integrated development environment with similar platform/language on client and web tiers
- C. Want to enhance client data access through closer coordination between client tier JavaScript and web tier objects

---

---

---

---

---

---

---

---

---

---

### Upshot.js



- Rebranded RIA/JS - WCF RIA Services for jQuery clients
- Part of Single Page Application project in MVC4
- Domain Service role taken by Controller which derives from DbDataController
- Client JavaScript code uses upshot.RemoteDataSource object

---

---

---

---

---

---

---

---

---

---

## Upshot.js



- Example of a view model (see BigShelf demo):

```
BigShelf.CatalogViewModel = function (options) {
  var self = this;
  var pageSize = 6;
  var booksDataSourceParameters = {};
  var booksDataSource = new upshot.RemoteDataSource({
    providerParameters: { url: options.serviceUrl, operationName:
      "GetBooksForSearch", operationParameters: booksDataSourceParameters },
    mapping: function (data) { return new BigShelf.Book(data,
      self.flaggedBooks) }, entityType: BigShelf.Book.Type
  });
};
```

- Controller metadata fed into client on server by Html helper before view is rendered to browser

```
upshot.metadata( @(Html.Metadata<BigShelf.Controllers.BigShelfController>());
```

---

---

---

---

---

---

---

---

---

---

## Upshot.js



- Analogous to WCF RIA Services in Silverlight
- Can apply sorting, filtering, etc. from client code
- Can track changes and do batched editing to reduce network calls to server
- Looks like a powerful example of the web remoting approach

---

---

---

---

---

---

---

---

---

---

## Demo



- <http://code.msdn.microsoft.com/ASPNET-Single-Page-d6d7172c>

---

---

---

---

---

---

---

---

---

---

### What's next?



- A look at HTML 5 and what it offers for RIAs

---

---

---

---

---

---

---

---

