An Overview of and Criteria for the Differentiation and Evaluation of RIA Architectures

Marcel Linnenfelser
Synflag Web Engineering, Germany

Sebastian Weber
Fraunhofer Institute for Experimental Software Engineering (IESE), Germany

Jörg Rech
Fraunhofer Institute for Experimental Software Engineering (IESE), Germany

ABSTRACT

An important aspect of Web 2.0 mentioned by Tim O'Reilly is the Rich User Experience. Web 2.0 applications offer the user a desktop-like interface to bring back efficiency and productivity. The Click-Wait-and-Refresh-Cycle of normal Web applications leads to a less responsive, and thus less efficient, user interface. To serve the needs of these so-called Rich Internet Applications (RIA), many different approaches have emerged, based either on Web standards or on proprietary approaches. This chapter aims at defining a qualified criterion system for comparing RIA platforms. Thereafter, those RIA platforms are selected and analyzed in terms of the criterion system that is most likely to become widely accepted.

KEYWORDS

Web 2.0, Rich Internet Application (RIA), RIA platforms, criterion system, overview, comparison, DHTML, Silverlight, Flex, JavaFX Script, GWT

INTRODUCTION

In his essay "What Is Web 2.0", Tim O'Reilly (2005) collected attributes that qualify a Web platform as Web 2.0. The key features of Web 2.0 platforms from the technological point of view are: User Generated Content, Tagging / Folksonomy, Content Syndication, and Rich User Experience. While the other features mentioned affect only some minor parts of the technological side of a Web platform, the Rich User Experience requires a fundamental architectural decision. This chapter will focus on Web X.0 technologies that enable a Rich User Experience and state several criteria for the differentiation and evaluation of these technologies for Web 2.0 services.

Web applications themselves have many advantages in software distribution and deployment. But as Kevin Hakman (2006) from TIBCO Software Inc. showed by changing over from a fat client to a Web client in Siebel Systems software in 2002, Web clients may have a serious impact on productivity. At one call center, there was a 30% productivity loss caused by the Click-Wait-and-Refresh-Cycle, as he called it. RIA technologies enable Web clients to measure up to fat clients regarding GUI usability, thus combining the advantages of Web clients and fat clients.

This chapter aims at providing a criterion system for evaluating RIA platforms and frameworks, which is defined in section "Definition of a criterion system". It is designed as a tool for decision makers to compare such platforms and to help them select an appropriate one for a specific project. The section "Platform Outlines" applies the criterion system to evaluate and compare those currently available platforms that are most likely to become widely accepted.

DEFINITIONS

Web application A *Web application* is an application "accessed over the World Wide Web by using a Web browser" ("WHATWG FAQ").

Rich Internet Application (RIA) The term was coined by Macromedia in 2002. "Macromedia defines RIAs as combining the best user interface functionality of desktop software applications with the broad reach and low-cost deployment of Web applications and the best of interactive, multimedia communication. The end result: an application providing a more intuitive, responsive, and effective user experience. Specifically, the best of the desktop includes providing an interactive user interface for validation and formatting, fast interface response times with no page refresh, common user interface behaviors such as drag-and-drop and the ability to work online and offline. The best of the Web includes capabilities such as instant deployment, cross-platform availability, the use of progressive download for retrieving content and data, the magazine-like layout of Web pages and leveraging widely adopted Internet standards. The best of communication means incorporating two-way interactive audio and video" (Duhl, 2003).

Asynchronous JavaScript + XML (AJAX) "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;
- asynchronous data retrieval using XMLHttpRequest;
- and JavaScript binding everything together" (Garret, 2005).

Offline Web application *Offline Web Applications* utilize Web technologies to build desktop applications. To enable the applications to run online and offline, they have to be granted file access in order to be able to save states ("TR10: Offline Web Applications").

RIA runtime environment An *RIA runtime environment* provides an environment that allows running platform independent RIAs. Usually, the runtime environment is available for different operating systems.

RIA framework The term *RIA framework* is used in this text to describe an application framework that supports the development of RIAs for one or more RIA runtime environments.

BACKGROUND

Today RIA is a hyped topic, causing many companies to enter the market of RIA platforms and frameworks. There exist numerous JavaScript-based frameworks enabling the development of RIAs that run directly in the browser, without the need for additional plug-ins. Adobe offers a framework called Flex, which enables the development of applications with desktop-like interfaces targeting the Flash player. Even Microsoft and Sun have come up with their own solutions for RIAs, but so far, only Microsoft has released a final version. And there are a lot more frameworks and platforms for building RIAs, e.g., Lobo[1], Curl[2], Omnis[3], Mozilla Prisma[4], to name but a few. The large number of competitors in the RIA platform and framework market makes it difficult to gain a general overview.

Douglas Engelbart presented the concept of an "oNLine System" (NLS) in December 1968. The system allowed two people to communicate via audio and video and to collaborate on a shared screen to create and edit text documents (Norman, 2005, p. 41).

In 1969, a team led by Leonard Kleinrock established a connection between two host computers over a network switch (Norman, 2005, p. 41). The first network was established.

Terminals such as DECwriter II and later DEC VT-100 made the first distributed applications

possible, at least from the users' point of view. In the 1970s, the Personal Computer (PC) appeared, and after 1985, computer networks fanned out (Ceruzzi, 1998, p. 6). Personal Computers allowed the use of fat client network applications. While being normal desktop applications, those fat clients could be as powerful as any other desktop application. The disadvantage of such applications is the need to install them on all client computers.

The Web was invented by Tim Berners-Lee in 1989-1990. He created the HTML and implemented the first HTTP server and the first browser (Norman, 2005, p. 100-102). Web applications are lightweight and do not need to be installed on the client computer. But as mentioned in the introduction, traditional Web applications may have a serious impact on productivity. With the release of Internet Explorer 5 in 1999, Microsoft introduced XMLHTTP ActiveX Control ("About Native XMLHTTP"), which allowed a type of applications later known as AJAX-based RIAs (Garret, 2005).

The term RIA was coined by Macromedia in 2002 to describe desktop-like Flash-based applications. In 2002, Macromedia released Flash MX featuring Flash UI Components (Mook, 2003).

In the following, a criterion system will be defined that allows characterization of RIA platforms, and thus comparison of the different RIA platforms. The outline of characteristics given in this chapter will help to evaluate the technologies regarding specific needs.

CLIENT-SERVER CROSS-SECTION

RIAs are basically client-server applications. They differ in the amount of processing done on the server and the client. For this examination, a layer model with three layers is assumed. The first layer hosts the data access, the second the business logic, and the last one the presentation logic.

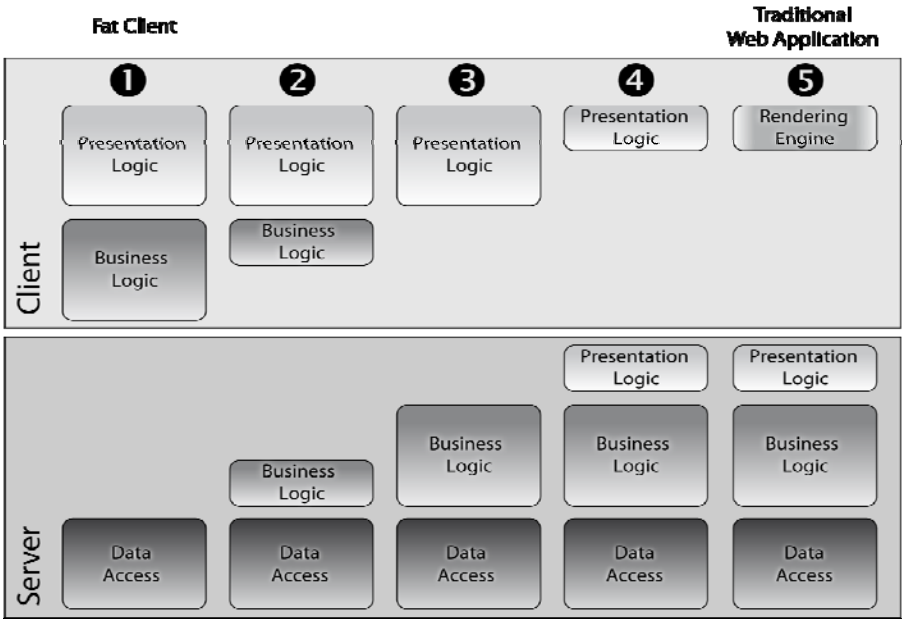

Figure 1 Cross-section through client-server architecture

Figure 1 shows five approaches to allocating the layers on client and server. Approach one shows a Fat Client (Mahemmoff, 2006, page 317), where all the processing takes place on the client side. Communication with the server is only necessary for manipulating or retrieving data. Data may be cached on the client. Approach two hosts only a part of the business logic on the client and the other part on the server, while approach three only performs retrieval and presentation on the client. To update the presentation in approach three, data has to be sent to the

server. After the data has been processed by the business logic, the results are forwarded to the client, where the presentation is then calculated. While approach four splits the presentation logic and calculates a part of the calculation on the server side, the last approach calculates the presentation on the server and sends a description (e.g., HTML) of it to the client, where the presentation is rendered following the description.

The first three designs may be implemented with any of the discussed RIA technologies. Approach four has to have some kind of rendering engine on the client side as well as a processing engine. In an example implementation of approach four, implemented using AJAX, the server-side presentation logic would process data obtained from the business logic and would generate HTML and JavaScript code to be sent to the client. Client-side events are handled by the generated JavaScript code. The server is requested via XMLHttpRequest (see section "AJAX"). The server-side processing logic generates HTML snippets to replace the HTML code of the updated sections of the page. Client-side JavaScript is used to replace the obsolete HTML.

Approach five is a traditional Web application where all the HTML is generated on the server. The HTML is sent to the client and rendered by the rendering engine. On every client-side event, a new request is sent and the whole page is refreshed.

DEFINITION OF A CRITERION SYSTEM

In the following, a criterion system is defined that allows evaluating and comparing RIA platforms. The criteria cover development aspects as well as runtime and availability issues.

Tool support: *Available development tools like debugger and profiler.*

Tool support is important for efficiency in development and testing.

Availability: *Availability on different operating systems.*

Depending on the audience, the RIA platform used has to support the target operating systems.

Available APIs and functionalities: *Available APIs such as collections, threading, and special language extensions.*

The availability of a functionality or type of API such as a 3D API may enable or disable a platform or framework for a certain project.

Language characteristics: *Characteristics like object-orientation, inheritance, and such.*

Language characteristics can determine the applicability of a platform or framework regarding the size of a software project. For example, use of an un-typed language may be inappropriate for large systems.

Runtime environment: *Characteristics of the runtime environment.*

The runtime environment determines the features, the platform independence, and the performance of a platform.

Extension challenges: *Challenges in creating and modifying UI elements.*

The RIA platforms differ in architecture. This leads to various extendibility challenges and different flexibility of the styling capabilities. For all-audience applications, styling capabilities may be essential. For some frameworks, extended tool support exists to assist designers.

Market penetration: *Market penetration regarding all Internet-connected client computers.*

Market penetration is important especially for all-audience applications.

Beyond the browser: *Many platforms allow developing applications to be deployed not only in the browser.*

Offline Web applications close the gap between desktop applications and Web applications. Applications with direct access to the file system can offer improvements in

terms of usability.

Interoperability: *Interoperability with JavaScript and thereby interoperability with other plug-ins.*

For some applications, a mix of the browser's native RIA capabilities and plug-ins is the best choice. Therefore, interoperability may be important.

Separation of design and logic: *Special language features and approaches to separate design (UI) and logic.*

The separation of design and logic can help to increase maintainability, but also makes it possible to split responsibilities between programmers and designers.

Supported media types: *Supported video and audio formats, as well as bitmap and vector graphic formats.*

The supported media formats may determine the decision for a specific RIA platform.

Installation: *Download size, install experience, and such.*

The download size and simplicity of installation of the runtime environment of a chosen framework may be irrelevant for business applications if automatic update mechanisms are in use at the customer's site. But at least for all-audience applications, these parameters are critical in terms of acceptance.

Supported devices: *Supported input and output devices.*

Access to webcams and microphones enables different kinds of applications, such as collaborative applications and applications that require taking cam shots (e.g., barcode reader via webcam).

The criterion system defined in this section will be used in the following section "Platform Outlines" to evaluate and compare the four platforms and frameworks AJAX, Microsoft Silverlight, Adobe Flex, and JavaFX.

PLATFORM OUTLINES

AJAX

The AJAX platform is made up of Web browser applications. It is based on the W3C and ECMA standards HTML, CSS, XML, JavaScript / ECMA Script (including JSON), and XMLHttpRequest-Object and similar mechanisms. Due to the multitude of existing browsers, the platform is highly heterogeneous. The browsers differ in their support of the standards. Thus, cross-browser compatiblity is an important subject for the AJAX platform and is sometimes hard to deal with.

Tool Support

Countless text editors are available with syntax coloring capabilities. Editors like Adobe GoLive and Adobe Dreamweaver allow WYSIWYG editing of HTML. These products have been used by designers for years, so there is much expertise available in the designer community. The programs support developers in creating HTML, CSS, and JavaScript, and feature tools for dealing with browser incompatibilities.

Several IDEs provide support for JavaScript. One of the most feature-rich IDEs for JavaScript is the Eclipse-based Aptana Studio[5]. Aptana comes with syntax coloring, code assist for HTML, CSS, and JavaScript, as well as JavaScript debugging for Firefox and Internet Explorer (IE). Debugging supports breakpoints and watched variables.

Firebug is a debugger extension for Firefox. It allows setting break points, analyzing the network traffic of AJAX applications, inspecting the page structure, and profiling JavaScript code.

The number of JavaScript frameworks available makes it a very time-consuming task to get an

overview. Also, the number of frameworks that provide JavaScript widgets[6] is anything but small and includes Dojo Dijit, Backbase, TIBCO General Interface, Ext JS, and Adobe Spry, to name but a few. Because there is not one single standard framework with one standard set of widgets, the only way of providing WYSIWYG editing of GUIs is for each framework to provide its own tools for this purpose. The GUI builders of TIBCO and Ext JS are written using their respective frameworks and run in the browser.

An interesting approach is the Google Web Toolkit (GWT). GWT uses a Java-to-JavaScript translator to add an abstraction layer. This allows using Java tools and the GWT development tools, including a full-featured debugger.

Availability

HTML, CSS, and JavaScript are available in nearly every browser in use, but many of the installed browsers are not fully standards-compatible. Bugs also induce non-standard-conformant or even unpredictable behavior[7]. The heterogeneity of the AJAX platform leads to increased development and testing effort.

More than 98% of installed browsers are AJAX-capable (“Browserwatch”) (“The Counter.com”) (at least in Germany), which means they support XMLHttpRequest (“The XMLHttpRequest Object”) objects or XMLHTTP ActiveX control (“About Native XMLHTTP”).

Available APIs, libraries, and functionalities

Designed to manipulate an HTML page after it is loaded by the browser, the key API of JavaScript is the Document Object Model (DOM). The DOM grants access to the elements of the currently loaded HTML page and to the attributes of the page elements, by providing methods and properties to retrieve, modify, update, and delete parts of the document (“The DOM and JavaScript”). The DOM has been standardized by the W3C (“Document Object Model (DOM)”) to provide a language-neutral and compatible interface for accessing content structure and style of documents. Manipulation of the HTML DOM using JavaScript is often referred to as Dynamic HTML (DHTML).

JavaScript provides support for working with arrays, doing calculations, and working with regular expressions.

Several libraries including Dojo Toolkit and Prototype[8] address problems with JavaScript version and browser incompatibilities by building abstraction layers and extending the browser DOM.

With the help of the Dojo Toolkit, JavaScript supports a technique called comet (“Comet: Low Latency Data for the Browser”). Comet uses long-lived HTTP connections to allow the server to push data to the client.

Language characteristics

A Web developer who develops for the AJAX platform should at least know three languages for doing client-side web development: HTML, JavaScript, and Cascading Style Sheets (CSS).

JavaScript is an object-based language based on the ECM 262 specification (“ECMAScript Language Specification 3rd Edition”).

JavaScript’s functional language features allow functional programming, which may lead to more compact code. For more information on functional programming, see (“Functional programming in the real world”).

Some language features like E4X (“ECMAScript for XML (E4X) Specification”) and XSLT are not available on all browsers.

CSS is used to style HTML elements of a Web page by applying properties.

Runtime environment

The runtime environment of HTML- and JavaScript-based applications consists of the rendering engine and the JavaScript interpreter. The rendering engine renders the visual representation of the HTML and CSS code and provides access to the DOM. The JavaScript interpreter parses and interprets the embedded JavaScript code and accesses the DOM provided by the rendering engine to manipulate the page display.

The runtime is heterogeneous. There are serious differences between the available browsers, even between versions of the same browser (“Microsoft's Interoperability Principles and IE8”).

Extension challenges

Describing how to extend the many different frameworks would go beyond the scope of this chapter. The one challenge common to all the extensions of the various frameworks is the heterogeneous runtime environment mentioned before. Creating widgets requires writing HTML, CSS, and JavaScript code for and testing of all the browsers to be supported.

Market penetration

Most, if not all Internet-connected client computers have support for HTML, CSS, and JavaScript, because all common client operating systems[9] include a Web browser. Most installed browsers are AJAX enabled; thus, they allow development of RIAs.

Beyond the browser

Since Internet Explorer 4.0, Microsoft has supported HTML Applications (HTA). HTAs are run like every other executable on Windows; thus, they have access to the file system and other privileges. The available platform features depend on which version of Internet Explorer is available (“Introduction to HTML Applications (HTAs)”).

Adobe has developed a runtime to build RIAs that deploy to the desktop. It is called AIR and supports development with HTML and JavaScript as well as with Flash and Flex. Unlike a browser or the Flash plug-in, AIR grants executed applications access to the file system.

Interoperability

Most plug-ins that can be embedded in Web pages can be accessed using JavaScript, and vice versa. For example, Java enables accessing JavaScript functions and the DOM (“Java-to-Javascript Communication”). A developer can also define methods that can be called from JavaScript (“JavaScript-to-Java Communication (Scripting)”). The External Interface class of the Flex framework allows Flex applications to access any JavaScript function and JavaScript to access defined Actionscript functions. Silverlight has a similar functionality.

Separation of design and logic

Usually JavaScript and HTML are mixed together in one file and JavaScript code is written directly into the event attributes of the tags. While the described code layout can cause readability problems, it is possible to assign all event handler functions without any JavaScript code in the HTML. Libraries like jQuery[10] and Dojo behavior[11] make it easier to access DOM nodes and assign event handler functions.

Frameworks often use some kind of templating mechanism to separate JavaScript code from HTML code. Dojo, for instance, uses templates to separate the implementation of widgets from the HTML code.

Supported media types

Normally, the only media types directly supported by browsers are images in the formats GIF, JPEG, and PNG[12]. Safari and Firefox also support a subset of Scalable Vector Graphics (SVG), an XML vector format. Internet Explorer supports VML, which was mentioned above. Other media types are supported through plug-ins, if available for the particular operating system and browser.

Installation

Browsers are installed with all relevant desktop systems. Microsoft Windows comes with Microsoft Internet Explorer, Mac OS X comes with Safari, and the available Linux distributions mostly come with Firefox and / or other browsers with the Mozilla Gecko rendering engine.

Supported devices

Generally, no special devices are supported without special plug-ins being installed.

Silverlight

Silverlight offers a browser plug-in based on the Windows Presentation Foundation (WPF). This text will mainly focus on aspects only found in Silverlight 2.

Tool Support

XAML (eXtensible Application Markup Language) is unlikely to be written by hand, but will be created by specialized software. Microsoft Visual Studio helps to create basic user interfaces with XAML from a developer's point of view. Applying a sophisticated design should be done by designers using Microsoft Expression Blend. Thus, Silverlight and XAML allow separating design and logic, as well as giving developers or designers an environment they are used to (MacDonald, 2007, page 22).

Availability

Microsoft provides a Silverlight player for Windows and Mac OS in Version 1.0 and 2.0 beta (as of March 19, 2008). The Mono project[13], which is supported by Novell, is developing a compatible open source alternative to the Silverlight player, called Moonlight[14].

Microsoft presented Silverlight 1.0 for Mobile on MIX08, which is restricted to JavaScript-based Silverlight 1.0 content. Nokia has announced Silverlight support on S60 on Symbian OS and Series 40 ("Nokia to bring Microsoft Silverlight powered experiences to millions of mobile users").

Neither the free SDK for one of the two Silverlight versions, nor the tools Visual Studio and Expression Blend are available for any platform other than Windows. This means that developers and designers are tied to the Windows platform, at least at the moment. The Mono project is planning to integrate an XAML designer into MonoDevelop, the Integrated Development Environment of the Mono project, which is based on Alan McGovern's Lunar Eclipse ("MonoTorrent").

Available APIs, libraries, and functionalities

Silverlight comes with a base class library, which is a compatible subset of the full .NET framework that includes collections, IO, generics, threading, globalization, XML, local storage, cryptographic services, libraries for the definition of global methods and types, generation of assemblies at runtime, events and delegates, and more (Guthrie, 2008) ("Common Language Runtime and Base Class Library in Silverlight").

Many of the functionalities mentioned above are known from other languages like Java. Silverlight libraries allow accessing a so-called isolated storage in which a partial trust application can store files. Thus, an application can store caches and user settings on the user's machine. Another major feature is the .NET Language-Integrated Query (LINQ) ("LINQ: .NET Language-Integrated Query"), which turns out to be a general-purpose query language extension of the C# language. LINQ looks much like SQL. It is used to query XML as well as other data.

The 3D namespace of the .NET framework 3.5 is missing in Silverlight's base class library.

Silverlight features rich network support, including support for calling REST, SOAP, POX, RSS, and standard HTTP services. Cross-domain network access and networking sockets are also included (Guthrie, 2008).

Language characteristics

As of Silverlight version 2.0, the Common Language Runtime (CLR)[15] is included. This allows using every language supported by the .NET framework with Silverlight, including C#, Python, and Ruby. Since C# is Microsoft's preferred language for the CLR, this text will focus on it.

C# was developed by Microsoft and has been standardized by the ECMA ("C# Language Specification (ECMA-334 4th Edition)") and the ISO ("ISO/IEC 23270:2003"). It is a high-level language similar to Java. Other useful language features are LINQ, delegates, enums, structs, and generics.

User interfaces of Silverlight applications are defined using a language called eXtensible Application Markup Language (XAML[16]). Because of the hierarchical nature of XML, an XML-based language is a good choice for defining GUI component trees.

Runtime environment

The runtime environment of Silverlight is Microsoft's CLR, an implementation of the Common Language Infrastructure (CLI) ("Common Language Infrastructure (CLI) (ECMA-335 4th Edition)"). The CLI defines an infrastructure that is able to execute multiple high-level languages. The languages are compiled into the Common Intermediate Language (CIL), the instruction set understood by the Virtual Execution System (VES). The infrastructure allows assemblies to run without modification on every platform the infrastructure is available on. In the managed environment, a garbage collector does automatic memory management. To increase execution speed, the infrastructure includes a Just-in-Time (JIT) compiler.

Extension challenges

Silverlight and the technologies used are well structured; thus, it is relatively easy to create custom UI controls. A ControlTemplate written in XAML defines the user interface of the control. The event handling and controller logic can be embedded into the XAML file, but placing it into a partial class is far more readable and maintainable ("Creating Custom Controls for Silverlight").

Unlike HTML and JavaScript, Silverlight applications are executed on a homogeneous platform; thus, testing a large number of alternative runtime environments is not necessary. The only compatibility problems that might occur affect Moonlight, the upcoming open-source implementation of Microsoft Silverlight.

Market penetration

There are no official numbers from Microsoft at the time of this writing, but the demand for Silverlight developers is low (Lai, 2008), meaning that only few companies are creating Silverlight content at all.

Beyond the browser

There is no desktop runtime environment available for Silverlight, except for the complete .NET framework, which is only available on the Windows platform.

Interoperability

Silverlight allows accessing so-called managed code[17] from JavaScript and vice versa. A developer is able to access properties and methods of managed code from JavaScript and to connect managed methods with JavaScript events. On the other hand, managed code can access the DOM and the access properties and methods of the DOM. JavaScript functions can be connected to managed events. JavaScript may also be used to access other plug-ins from Silverlight embedded in the same page ("How to: Call Managed Code from JavaScript") ("Accessing the HTML DOM from Managed Code").

Separation of design and logic

Partial classes allow splitting a class and prorating it over several files. Since an XAML file is usually translated into a class, a partial class can be used to add methods to the generated class. This way, no programming code has to be in XAML files. Event handlers for certain events can be specified with the corresponding attribute, e.g., the Click attribute is given the name of the method for handling the click event, which is defined in the partial class.

Supported media types

The Microsoft Silverlight plug-in has built-in support for various media formats. The Windows Media Audio (WMA) format is supported, as is MP3 audio. The Silverlight plug-in also supports the WMV7-9 video codecs. WMV9 is Microsoft's implementation of the standard VC-1 codec. VC-1 codec enables 720p HD Movies. Progressive downloading[18] and streaming[19] are also supported.

Installation

The Silverlight version 2.0 (currently beta as of March 18, 2008) runtime for Windows has a file size of 4.4 MB. A developer embedding a Silverlight movie into a Web page is able to provide an attribute called 'pluginspage' to point to a download location for Microsoft Silverlight. Microsoft does some player detection using JavaScript, which embeds the movie if the player is detected[20].

Supported devices

Supported devices are unknown at the time of writing. See ("Silverlight FAQ") for up-to-date information.

Flex

Flex is a framework originated by Adobe (formerly Macromedia) to enable a more developer-like approach to creating Flash-based applications than Flash CS Professional, which aims at designers wanting to create animated Web sites and other animated content.

Tool Support

Since Flex is an Adobe product, Flex Builder provides the most extensive support for Flex development, such as syntax highlighting, code-assist, life-error highlighting, refactoring, debugging, and profiling. Flex Builder also includes a GUI designer for visually creating Flex-based GUIs. The GUI designer creates MXML code, an XML-based language used to define UI

component trees. Source code editors are available for MXML and Actionscript 3.0, the language Flex is based on. Another IDE with Flex support is IntelliJ IDEA, but it can only be used as an editor, including code-assist and syntax highlighting ("IntelliJ IDEA, JavaScript Editor").

In addition to Flex Builder, Adobe offers a large lineup of tools for changing the appearance of and designing new Flex components. Since Flex 3, Adobe has offered the Flex Skin Design Extensions for Fireworks CS3, Flash CS3 Professional, Illustrator CS3, and Photoshop CS3, which allow creating skins for Flex components that offer more options for changing the visual appearance of components than styles[21]. For Flash CS3 Professional, Adobe offers an extension called Flex Component Kit. The kit allows creating Flex components with Flash CS3 Professional.

Currently, Adobe is working on a new software similar to Microsoft Expression Blend, code-named Thermo ("Thermo").

The Flex SDK is available as open source under the Mozilla Public License (MPL). It features a compiler and a debugger for MXML and Actionscript 3.0 and the Flex framework, as well as the core Actionscript libraries.

Availability

The Flex framework is based on the Adobe Flash player. For Flex versions 2 and 3, Flash player version 9 is the minimum requirement. Directly supported by Adobe are the following platforms ("Adobe Flash Player: System Requirements"): Windows 98, ME, 2000, 2003 Server, XP, Vista, Mac OS X 10.1.x to 10.4.x, Red Hat and SUSE Linux, Solaris 10. Although Flash Lite 3 is available for many cell phones, Flex version 2 and higher is not supported due to Flash Lite's restriction to Flash 8 content. RIAs for Flash Lite 3 can be developed using the Flash Professional authoring environment and Actionscript 2.0 instead of Actionscript 3.0.

The Flex 3 SDK is available for Windows 2000, 2003 Server, XP, Vista, Mac OS X 10.4.7-10.4.10 and 10.5[22], Red Hat, SUSE Linux, and Solaris 9 and 10[23]. The Adobe website provides different kinds of information for the Windows platform ("Adobe - Flex 3: System requirements").

Flex Builder 2 and 3, Adobe Fireworks CS3, Illustrator CS3 and Flash Professional CS3 are available for Windows and Mac OS X only. Since Flex Builder is based on Eclipse and the SDK is available for Linux, it should not be a problem for Adobe to provide a Linux version in the future.

Available APIs and functionalities

Besides a rich pool of UI controls, Flex comes with the functionalities of Actionscript 3.0, the Flex API, and Flash Player API. Actionscript 3.0 features an XML language extension called ECMAScript4XML (E4X) (Moock, 2007, page 353), which offers easy access to XML data and allows selecting XML elements. The Flash 2D display API allows dealing with interactive visual objects, bitmaps, and vector content (Mook, 2007, page 457). Further features are: Animation API (Mook, 2007, page 610), effects and transitions (Kazoun and Lott, 2007, page 232), back-button handling[24], data bindings (Kazoun and Lott, 2007, page 268), RPC APIs[25] and sockets, validation and formatting (Kazoun and Lott, 2007, page 288), and loading external content (Mook, 2007, page 762).

Flash player allows an SWF file to store data locally on the user's computer. The default maximum data that can be stored is 100 kb, but the user can agree to store more ("Flash Player Help", "Local Storage Settings").

Language characteristics

Because Actionscript 3.0 (AS 3.0) follows the ECMAScript Edition 4 specification, which is currently under development, most of the statements on JavaScript in section "AJAX" are correct in this case, too. But AS 3.0 has learned the advanced features of modern object-oriented languages. AS 3.0 features single inheritance, interfaces, data types, namespaces, metadata, and exception handling. Although AS 3.0 is a typed language, it still has the dynamic abilities of ECMAScript Edition 3. A variable can be defined without a type[26] or with a wildcard.

AS 3.0 allows adding instance variables and instance methods at runtime (Mook, 2007, page 279). Something similar to Ruby's mixins ("Programming Ruby") is also possible ("Ruby-like Mixins in Actionscript 3.0"). Using interfaces makes the AS 3.0 "mixins" type-safe.

E4X extends Actionscript 3.0 with XML. It allows using XML directly in the source code and provides convenient handling of XML data (Mook, 2007, page 353).

MXML, the markup language of Flex, is an alternate way of defining a class. It allows defining component trees declaratively (Kazoun and Lott, 2007, page 43).

Actionscript features garbage collection, like most modern languages that target a virtual machine.

Runtime environment

The runtime environment of Flex is Flash player 9 and higher. Flash player includes the Display API (Mook, 2007, page 457), among other things. Flash player also includes two virtual machines, but only the second one – the AVM2 – can run Flex 2 and 3 applications.

Actionscript 3.0 bytecode runs in the new AVM2 ("Adobe Flash Player: Features") virtual machine included in Flash player 9 and above and Adobe AIR 1.0 and above. AVM2 is open-sourced under the name Tamarin ("Tamarin Project"). It features a Just-in-Time (JIT) compiler ("Adobe/Mozilla Tamarin Project Frequently Asked Questions") to increase execution speed by creating native code for a particular hardware platform.

Extension challenges

A custom Flex component is a subclass of UIComponent. Custom components are created using MXML or Actionscript. Subclasses of container components with children are called composite components. If a component is instantiated using an MXML tag, the attributes can specify events, styles, and values of properties.

As with Silverlight, the applications are executed in a homogeneous environment. Some small problems seem to exist. For example, Flash player on Mac OS X is not able to handle the mouse wheel[27].

Market penetration

Since Flex 2 and Flex 3 need, as a minimum, Flash player 9, only the market penetration of version 9 and above matters. At the time of this writing, Flash player 9 is the latest version available and has a market penetration of 95.7% in the mature markets[28] and 93.3% in the emerging markets[29] ("Adobe Flash Player Version Penetration").

Beyond the browser

Since the release of AIR 1.0 on February 25, 2008, a desktop runtime environment for Flex applications has been available. Flex on AIR has some additional functionalities. Besides the Flash player, Adobe also packaged the WebKit rendering engine from Apple's Safari browser. This makes an HTML component available to Flex, which has the complete rendering capability of a modern rendering engine. Also, AJAX-based applications can run in AIR, and can also be

combined with Flex applications. Applications running in AIR are allowed to access the local file system ("Adobe AIR Local File System Access") and enable drag and drop from the desktop or other programs ("Flickr Floater").

Interoperability

Flex offers three ways of data communications on the client: local connections, shared objects, and external interface. Local connections allow .swf files to communicate as long as they are running on the same machine, no matter in what environment. Shared objects allow storing locally shared objects on the client, which can be loaded the next time the application is running. The data is stored in the meantime. Finally, the external interface allows accessing the .swf file from the host environment and vice versa. In the case of the Flash player running in a browser, the external interface allows interacting with JavaScript (Kazoun and Lott, 2007, page 355).

Separation of design and logic

An MXML file is translated into a class and mx:Script allows embedding Actionscript code into an MXML file, such as methods and properties. The easiest way to separate MXML and Actionscript code is to use the MXML counterpart of the reserved word include, mx:Script with the attribute source to specify the Actionscript file to be included. This is an approach similar to C#'s partial classes used in conjunction with an XAML template.

Another, but more complex, way to separate MXML and Actionscript is to use Adobe's framework Cairngorm. Cairngorm is called a micro-architecture and provides an Actionscript-like implementation of J2EE blueprint patterns ("J2EE Patterns Catalog"), like the Front Controller and the Business Delegate.

Supported media types

Flash player supports various video codecs up to HD. It supports H.263 playback and encoding, H.264, and On2 VP6 playback. The supported audio formats are MP3 and HE-AAC ("Datasheet Adobe Flash Player 9").

Additionally, Flash player supports PNG, JPG, and GIF formats for displaying bitmap images. SWF files can be embedded and loaded at runtime, but SVG files can be embedded with Flex at compile time only[30] ("Embedding Application Assets").

Installation

The Flash player's installation package of the latest Windows version 9,0,115,0 (on March 18, 2008) is 1.5 MB. Installation is very easy. Adobe AIR is a separate download and the file size is 11.2 MB for the latest version 1.0 (on March 18, 2008). The attribute pluginspage mentioned in the installation paragraph of Silverlight is available for Flash, too. Adobe offers the Flash Player Detection Kit, which includes Express Install. Express Install features a player-based installation process, which installs the Flash player and returns the user to the page that requested the plug-in ("Flash Player Detection Kit").

Adobe ships an interesting possibility for a seamless installation of AIR applications called badges. Badges allow installing an AIR application via an SWF embedded in a web page, regardless of whether AIR is installed or not. If AIR is missing on the system, it is automatically installed with the AIR application ("AIR Install Badges").

Supported devices

The Flash player supports audio output, mic input ("Flash Player Help", "Microphone Settings"), and video input through a camera ("Flash Player Help", "Camera Settings").

JavaFX

"The JavaFX family of products is based on Java technology, designed to simplify and speed the creation and deployment of high-impact content for a wide range of devices. JavaFX technology enables developers and designers to create and deploy consistent user experiences, from the web page to desktop to mobile device to set-top box to Blu-ray Disc" ("JavaFX Technology FAQs").

JavaFX comes with a new scripting language called JavaFX Script, which has a different syntax than Java. It can be executed in an interpreted mode, but may also be compiled directly to bytecode for the JVM.

Tool Support

Tool support for the Java programming language is very extensive, since many IDEs have extensive support. Some of the best known tools are Eclipse, NetBeans, and IntelliJ IDEA. JavaFX plug-ins are available for all of them. The JavaFXPad editor[31] and the NetBeans plug-in[32] feature realtime rendering of JavaFX Script. Since JavaFX is young and no stable release is available at the time of writing, the features of the plug-ins are limited.

Java support of the mentioned IDEs includes syntax highlighting, code-assist, life-error highlighting, refactoring, debugging, and GUI designing[33].

Currently, no design tools are available for JavaFX, except for the realtime rendering mentioned above. Sun is putting "a lot of effort into interoperability with Adobe tools", because designers know them well and have worked with them for years ("Sun's JavaFX tools to interop with Adobe").

Availability

Java itself is available for nearly every client and server operating system. Sun directly supports Windows, Linux, and Solaris ("System Requirements for JRE 6.0"). Apple delivers its own Java versions with Mac OS X; the latest version at the time of writing was Java Standard Edition (SE) 5.0. Java SE 6.0 for Mac OS X was available as a developer preview ("Java"). Since Java was released as open source in mid-2007, Java can be ported to any platform.

Java is also available on many mobile phones. Java Micro Edition (ME) is a slim version of Java SE suitable for the limitations of mobile devices ("Java ME at a Glance").

Sun has announced a new mobile operating system, called JavaFX Mobile, built on top of a Linux kernel and providing a built-in Java Virtual Machine (JVM) and a JavaFX environment ("JavaFX Mobile - Overview").

Java SE 5 or 6 is needed for JavaFX Script development.

JavaFX technology is planned to be made available for the Java ME profiles Connected Limited Device Configuration (CLIC) and Mobile Information Device Profile (MIDP).

Available APIs and functionalities

JavaFX can access the complete class library of the host Java Runtime Environment (JRE), thus it depends on the JRE which APIs are available. On a Java SE, a wide range of APIs is available including networking, IO, security, cryptography, formatting, regular expressions, threading, and more.

Besides the bundled class library, many third-party libraries are available. For example, the Lobo Project offers a complete HTML rendering engine called Cobra[34], the Apache project offers a lot of libraries for XML processing and many other common tasks[35], and Java3D offers 3D capabilities[36].

Language characteristics

Java is a modern programming language. It is object-oriented and statically typed. Java features classes, interfaces, and exception handling. Memory management is done by a garbage collector.

JavaFX Script is a new scripting language targeting the JVM. Like Java, JavaFX Script is statically typed and object-oriented. Unlike Java, JavaFX offers multiple inheritance. Although complete programs may be written in JavaFX Script, the key concepts were designed with user interfaces, graphics, and animation in mind ("JavaFX != JavaFX Script"). Object literals are used to declaratively instantiate classes, which allows defining UI component trees in a readable manner. Object literals are used for the same purpose as XAML and MXML. The reserved word bind allows binding variables, attributes of objects, value expressions, or even return values of operations[37] to a certain attribute. This means that the latter is updated every time the bound value changes. Bindings allow connecting UI components declaratively. Instead of class constructors and getters and setters, JavaFX offers SQL-like triggers. Triggers are declared to fire on certain events like insertion, deletion, and replacement of data. For further information, see ("The JavaFX Script Programming Language").

Runtime environment

The runtime environment of Java includes the Java Virtual Machine (JVM), a stack-based virtual machine, and the class library. Currently, there are several editions available. Java Micro Edition (ME) features many different profiles and is dedicated to embedded and mobile devices. The Java Standard Edition is appropriate for desktop computers. Sun plans to drop Java ME in a few years in favor for Java SE, because mobile devices are getting enough power ("Sun starts bidding adieu to mobile-specific Java").

The virtual machine runs so-called Java bytecode. Unlike Flash player's AVM2 and Microsoft's CLR, the JVM is a HotSpot VM ("Java SE Hotspot at a Glace"). A HotSpot VM identifies code "worthy" of being optimized and compiled into machine code, instead of compiling the whole application.

JavaFX applications are embedded into the browser using the Java plug-in. The special flavor of a Java application to run in the browser is called Java Applet.

Compiled JavaFX Script runs directly in the JVM, but JavaFX Script may also be executed by an interpreter, which is written in Java and can be embedded into Java programs.

Extension challenges

JavaFX Script user interface components inherit from Widget. Since UI components are normal JavaFX classes, modification of existing UI components is simply done by extending the components class using inheritance. Composite UI components are usually created by extending CompositeWidget, composite canvas components by extending CompositeNode. To build up the composite component, the object literal syntax may be used in the corresponding compose method.

Market penetration

Available numbers differ. Adobe sees Java on 84.6% ("Flash Player Penetration") of all Internet-enabled desktops in mature markets[38]. Danny Coward states that 91% of all PCs were running Java in June 2007. Six months after the release of Java SE 6, 13% of all PCs were running Java SE 1.6, according to him (Coward, 2007, slide 11).

Beyond the browser

Java SE features Java Web Start, which enables deployment of standalone applications over

networks with a single click. A Java Network Launch Protocol (JNLP) file specifies the files to be downloaded and the main class. After downloading is finished, the application is started immediately.

Interoperability

The Java browser plug-in provides an easy way to access the DOM of the embedding Web page and to call JavaScript functions ("Java-to-JavaScript Communication"). It is also possible to access properties and methods of applets ("JavaScript-to-Java Communication").

Separation of design and logic

JavaFX Script does not differ that much from Java when it comes to action listeners. In JavaFX, a function is used as an action listener, instead of the usually used inner classes of Java. The strategies for separating the model from the view are the same in both languages. An example strategy could be to reduce the code in the action operations to a single call of a method of the model.

Supported media types

With JMF, Java applications and Applets can playback video and audio. While JMF supports a wide range of codecs, it still cannot compete with Flash and Silverlight, which both support high-quality movies up to HD with their modern codecs ("Java Media Framework API"). For version 1.0, extended support for high-quality audio and video is planned[39].

Installation

Sun offers a JavaScript library to facilitate the installation process if a user has no JRE or if the installed version is too old ("Deployment Toolkit").

The JRE is the heaviest download of all discussed technologies, with 15.18 MB for the multi-language Windows version. The developers from Sun refer to the installation process as slow and complicated and to the startup time as poor (Coward, 2007, slide 26). Sun plans to make the next version modular, and faster to start, and wants to improve the installation experience.

Supported devices

No out-of-the-box support for webcams and microphones is available. Audio output is possible.

PERFORMANCE TESTS

Alexey Gavrilov created a benchmark called Bubblemark[40], which offers implementations of the same animation for different platforms. The following were chosen for a comparison: DHTML, Silverlight with interpreted JavaScript (SL JS), Silverlight with Common Language Runtime (SL CLR), Flex running on Flash player 9, JavaFX Script interpreted (JFX), JavaFX Script optimized (JFX opti.), and Java Swing. The results of the Mac OS X benchmarks are shown in Table 1, the results of the Windows benchmarks in Table 2. The Mac test machine was an Apple MacBook Core Duo 2GHz running Mac OS X 10.4. The Windows test machine was a custom PC driven by an AMD 2500+ Barton with an ATI 9600 XT graphics card running Windows XP SP2. The test browser on Mac OS X was Safari 3.1 and IE 6 on Windows. Silverlight version 1.1 alpha was used, because the benchmark for Silverlight CLR was incompatible with Silverlight 2.0 beta. Java on Mac OS X was the bundled Java SE 5.0; Windows had Java SE 6.0 installed.

Table 1 Bubblemark Mac OS X results

	DHTML	SL JS	SL CLR	Flex	JFX	JFX (opti.)	Java Swing
16	94 fps	86 fps	75 fps	55 fps	15 fps	64 fps	185 fps
32	88 fps	52 fps	44 fps	46 fps	6 fps	42 fps	184 fps
64	65 fps	26 fps	24 fps	29 fps	3 fps	22 fps	134 fps
128	25 fps	11 fps	12 fps	15 fps	< 1 fps	12 fps	82 fps

Table 2 Bubblemark Windows results

	DHTML	SL JS	SL CLR	Flex	JFX	JFX (opti.)	Java Swing
16	31 fps	54 fps	65 fps	38 fps	7 fps	27 fps	52 fps
32	15 fps	25 fps	35 fps	21 fps	3 fps	17 fps	30 fps
64	6 fps	10 fps	25 fps	9 fps	1 fps	10 fps	17 fps
128	3 fps	4 fps	18 fps	4 fps	< 1 fps	5 fps	9 fps

Unfortunately, there is no compiled JavaFX Script benchmark on the site, but the optimized version and the Java Swing version should provide an indication of how fast a compiled JavaFX version would be. The first thing to note is that the optimized JavaFX version reaches significantly higher frame rates than the one that is not optimized. The class that does calculations for collision detection is implemented as a Java class and therefore is compiled and not interpreted. The results of a completely compiled version should be somewhere between the optimized JavaFX Script benchmark and the Java Swing benchmark, since JavaFX Script's UI components are based on Swing and Java2D.

It is obvious that Apple highly optimized the JVM on Mac OS X. Compiled JavaFX should be a high-performance solution on Windows and Mac OS X. Safari's rendering engine WebKit also offers high performance. Adobe uses the same rendering engine in its AIR.

As expected, Silverlight with .NET CLR and JIT compiled assemblies is faster than the interpreted JavaScript Silverlight. Hence, it is all the more disappointing that Silverlight CLR performs so poorly on Mac OS X compared to Silverlight JavaScript.

Flex seems to be slow when the frame rates with 16 balls are compared, but if all results are taken into account, Flex becomes more competitive the more balls are displayed. But still, the results of Flex are disappointing compared with DHTML in Safari and Internet Explorer, if taking into account that Flash player 9 features a JIT compiler for Actionscript 3.0 assemblies.

CONCLUSION AND OUTLOOK

This chapter provided an overview of a sample of RIA technologies. A number of aspects have to be considered when the right technology for a certain project has to be chosen. First of all, the audience and the purpose of an application have to be determined. If the audience is the totality of all Internet users, then only those technologies can be taken into consideration that are available on enough client computers. If search engine marketing is important, naturally, plain HTML is the best solution, because search engines have been created with the specific characteristics of hypertext in mind. The use of AJAX techniques may cause a serious impact on the ability of search engines to index a website, because search engines do not evaluate JavaScript ("CSS, AJAX, Web 2.0 & Search Engines"). Although Google and other search engines are able to index Flash movies, usually the complete content is in one file. This prevents reasonable search engine optimization ("Get Flash Sites Ranked in Search Engines"). The dynamic contents of a Flash movie, Silverlight application, or Java Applet cannot be indexed by any search engine; thus, other techniques have to be applied ("Search enabling Silverlight and

AJAX web applications"). One aspect to consider are the skills of the available developers. Windows .NET developers are better off with Silverlight, for instance.

Another constraint of a project is the budget. Thus, it is important to minimize costs by choosing a technology that helps to achieve that goal. AJAX requires extensive testing on many different browsers on different operating systems, because of its heterogeneous platform. Especially on older browsers, it is likely to detect rendering problems. One advantage of AJAX is that it is available on nearly every Internet-connected desktop computer without the need to install a plug-in. Furthermore, plug-ins can be used to add additional functionalities and support for media types. The plug-in-based technologies are handicapped because they depend on a plug-in on the client computer. But if the appropriate plug-in is installed, a homogeneous platform is provided with only one environment to be tested. Silverlight is a new, but stable platform with good tool support. Currently, Silverlight is not widespread. JavaFX is also a relatively new platform. At the time of this writing, no stable release was available. A final release of JavaFX Desktop 1.0 is planned for the fall of 2008; JavaFX Mobile and TV 1.0 will ship in spring 2009 ("Sun offers JavaFX road map"). The existing development tools lack many features; design tools are not available, but a tool for converting SVG into JavaFX does exist ("JavaFX SVG Translator Preview").

At the time of this writing, the only plug-in-based platform with sufficient market penetration and good tool support is Flash / Flex. This makes it the platform of choice for plug-in-based applications aimed at the general public. At the moment, Silverlight is only suitable for applications where it is possible to make sure that the plug-in will be installed on all client computers. JavaFX is currently not ready for productive systems and should not be used until a stable release is available.

In the near future, offline Web applications will be widespread at least in companies. They offer features similar to desktop applications, but can be installed easily by accessing an URL. No administrator privileges are required for installation.

Today Adobe Flash and AJAX are the dominating RIA platforms. But the other two competitors JavaFX and Silverlight may be at the heels of Flash and AJAX in the near future, if the installation is hassle-free. While Java already has a widely installed basis, for JavaFX, an update to Java 6.0 will be necessary. Silverlight as a new platform has nearly no installed basis. However, if the installation process is easy and seamless and Sun and Microsoft can convince some big players in the Internet business to use their platform (such as Youtube, which is using Flash right now), the basis can grow very fast.

On mobile devices, Java is in the front. JavaME is installed on over 70% of today's mobile handsets[41]. The Flash Lite player is also widespread, but it is not sufficient for running Flex applications. Microsoft has a mobile runtime, too. As mentioned before, Nokia agreed to include this runtime in future handsets.

Current smart phone operating systems such as iPhone OS, Windows mobile, Google Android, and Symbian, one realizes that every one of them has its own incompatible programming model. RIAs on smart and other phones would help to reduce the costs of making an application available on all relevant mobile platforms. Only the runtime has to be ported.

Since today's phones offer more and more services, to enable a broader range of application types, the runtime should offer access to special phone features, such as GPS.

In our opinion, the mobile RIA market will become one of the most exciting markets for future RIAs. New packet-based data transfer technologies as well as current and future combined phone and data contracts will allow everybody to be "always on". The data rates have increased continuously during the last few years. Today's rates are higher than the home Internet access most people had a few years ago. The user experience of those applications is one of the major challenges for user acceptance. Apple's iPhone currently leads the way in mobile user interface design and interaction. Future RIAs will be able to utilize advanced types of input devices such

as multi-touch displays and to interpret user gestures.

To keep the cell phones handy, they have to stay rather small. Thus, the space on the screen is limited. RIAs have to deal with this constraint by introducing new ways of organizing user interfaces. Zooming the presentation, like Apple's iPhone Safari does, is no viable way for easily and effectively usable RIAs. One solution towards an optimal usage of the available screen space may be the context-sensitive rearrangement of the user interface. To help the user follow the transitions between the states of the user interface, these have to be animated. The concept of states and animateable transitions is already implemented in Adobe Flex.

Another issue where no standard solution exists is searchability. While Google and Yahoo have extended their indexers in order to be able to process Flash movies, this is insufficient for RIAs. Indexers can only work with content included in the indexed documents. This is a common problem of all RIA platforms. RIAs load the bulk of the content at runtime. Because search engine crawlers cannot navigate those applications, they are unable to index the content. In a project of Fraunhofer IESE called SOP (Software Organization Platform) 2.0 (Weber et al., 2008), we offer a hybrid user interface. A hybrid interface in terms of our implementation offers both an HTML and an Adobe Flex interface at the same time. Thus, search engines can crawl the HTML interface to explore the complete content. To enable the changeover from the HTML interface to the RIA interface, it must be possible to alter the state of the RIA by defining initialization parameters. If a user clicks on a search engine link, he is directed to the HTML document. The changeover to the RIA interface can proceed upon a trigger by the user or automatically.

The criterion system presented in this chapter was originally created in order to select an appropriate RIA platform for the SOP 2.0 project mentioned above.

REFERENCES

About Native XMLHTTP. Retrieved March 27, 2008, from http://msdn2.microsoft.com/en-us/library/ms537505.aspx

Accessing the HTML DOM from Managed Code. Retrieved March 28, 2008, from http://www.silverlight.net/Quickstarts/Dom/DomAccess.aspx

Accessing the HTML DOM from Managed Code. Retrieved March 31, 2008, from http://www.silverlight.net/Quickstarts/Dom/DomAccess.aspx

Adobe - Flex 3: System requirements. Retrieved March 28, 2008, from http://www.adobe.com/products/flex/systemreqs/

Adobe – Flash Lite. Retrieved March 28, 2008, from http://www.adobe.com/products/flashlite/

Adobe AIR Local File System Access. Retrieved March 28, 2008, from http://labs.adobe.com/wiki/index.php/AIR:Articles:Adobe_AIR_Local_File_System_Access

Adobe Flash Player : Features. Retrieved March 28, 2008, from http://www.adobe.com/products/flashplayer/productinfo/features/

Adobe Flash Player : System Requirements. Retrieved March 28, 2008, from http://www.adobe.com/products/flashplayer/productinfo/systemreqs/

Adobe Flash Player Version Penetration. Retrieved March 28, 2008, from http://www.adobe.com/products/player_census/flashplayer/version_penetration.html

Adobe/Mozilla Tamarin Project Frequently Asked Questions. Retrieved March 28, 2008, from http://www.mozilla.org/projects/tamarin/faq.html#avm2

AIR Install Badges. Retrieved March 28, 2008, from http://blogs.adobe.com/simplicity/2007/06/air_install_badges.html

Browserwatch. Retrieved March 27, 2008, from http://www.w3b.org/trends/browserwatch.html

C# Language Specification (ECMA-334 4th Edition). Available from http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf

Chafic Kazoun and Joey Lott. (2007). Programming Flex 2. O'Reilly

Colin Moock. (2003). What is a Flash MX Component? O'Reilly Web DevCenter.

Colin Moock. (2007). Essential Actionscript 3.0. O'Reilly

Comet: Low Latency Data for the Browser. Retrieved March 28, 2008, from http://alex.dojotoolkit.org/?%20p=545

Common Language Infrastructure (CLI) (ECMA-335 4th Edition). Available from http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-335.pdf

Common Language Runtime and Base Class Library in Silverlight. Retrieved March 28, 2008, from http://msdn2.microsoft.com/en-us/library/cc221412(vs.95).aspx

Creating custom controls for silverlight. Retrieved March 28, 2008, from http://www.silverlight.net/Quickstarts/BuildUi/CustomControl.aspx

Creating Custom Controls for Silverlight. Retrieved March 28, 2008, from http://www.silverlight.net/Quickstarts/BuildUi/CustomControl.aspx

CSS, AJAX, Web 2.0 & Search Engines. Retrieved March 28, 2008, from http://www.seroundtable.com/archives/006889.html

Danny Coward. (2007). What's new in java SE 7. In JAZOON07. Available from http://www.scribd.com/doc/257998/Whats-in-Java-SE-7

Datasheet Adobe Flash Player 9. Retrieved March 28, 2008, from http://www.adobe.com/products/flashplayer/pdfs/Datasheet_Flash_Player_9_ue.pdf

Deployment Toolkit. Retrieved March 28, 2008, from https://jdk6.dev.java.net/testDT.html

Document Object Model (DOM). Retrieved March 28, 2008, from http://www.w3.org/DOM/

ECMAScript for XML (E4X) Specification. ECMA. Available from http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-357.pdf

ECMAScript Language Specification 3rd Edition. ECMA. Available from http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf

Embedding Application Assets. Retrieved March 28, 2008, from http://www.adobe.com/devnet/flex/quickstart/embedding_assets/

Eric Lai. (2008). Little demand yet for silverlight programmers. Retrieved March 30, 2008, from http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9066838

Flash Player Detection Kit. Retrieved March 28, 2008, from http://www.adobe.com/products/flashplayer/download/detection_kit/

Flash Player Help. Retrieved March 28, 2008, from http://www.macromedia.com/support/documentation/en/flashplayer/help/index.html

Flash Player Penetration. Retrieved March 28, 2008, from http://www.adobe.com/products/player_census/flashplayer/

Flickr Floater. Retrieved March 28, 2008, from http://www.adobe.com/devnet/air/flex/articles/flickr_floater_04.html

Functional programming in the real world. Retrieved March 28, 2008, from http://homepages.inf.ed.ac.uk/wadler/realworld/

Get Flash Sites Ranked in Search Engines. Retrieved March 28, 2008, from http://www.clickz.com/showPage.html?page=3419561

How to: Call Managed Code from JavaScript. Retrieved March 28, 2008, from http://www.silverlight.net/quickstarts/Dom/ManagedCodeAccess.aspx

How to: Call Managed Code from JavaScript. Retrieved March 28, 2008, from http://www.silverlight.net/quickstarts/Dom/ManagedCodeAccess.aspx

IntelliJ IDEA, JavaScript Editor. Retrieved March 28, 2008, from http://www.jetbrains.com/idea/features/javascript_editor.html#flex

Introduction to HTML Applications (HTAs). Retrieved March 28, 2008, from http://msdn2.microsoft.com/en-us/library/ms536496(VS.85).aspx

Introduction to html applications. Retrieved March 28, 2008, from http://msdn2.microsoft.com/en-us/library/ms536496(VS.85).aspx

ISO/IEC 23270:2003. Retrieved March 28, 2008, from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=36768

J2EE Patterns Catalog. Retrieved March 28, 2008, from http://java.sun.com/blueprints/patterns/catalog.html

Java ME at a Glace. Retrieved March 28, 2008, from http://java.sun.com/javame/index.jsp

Java Media Framework API. Retrieved March 28, 2008, from http://java.sun.com/products/java-media/jmf/

Java SE Hotspot at a Glace. Retrieved March 28, 2008, from http://java.sun.com/javase/technologies/hotspot/

Java. Retrieved March 28, 2008, from http://developer.apple.com/java/

JavaFX != JavaFX Script. Retrieved March 28, 2008, from http://weblogs.java.net/blog/joshy/archive/2007/09/javafx_javafx_s.html

JavaFX Mobile - Overview. Retrieved March 28, 2008, from http://www.sun.com/software/javafx/mobile/index.jsp

JavaFX SVG Translator Preview. Retrieved March 28, 2008, from http://blogs.sun.com/chrisoliver/entry/javafx_svg_translator_preview

JavaFX Technology FAQs. Retrieved March 28, 2008, from http://java.sun.com/javafx/faqs.jsp

JavaScript-to-Java Communication (Scripting). Retrieved March 28, 2008, from http://java.sun.com/j2se/1.5.0/docs/guide/plugin/developer_guide/js_java.html

JavaScript-to-Java Communication. Retrieved March 28, 2008, from http://java.sun.com/j2se/1.5.0/docs/guide/plugin/developer_guide/js_java.html

Java-to-Javascript Communication. Retrieved March 28, 2008, from http://java.sun.com/j2se/1.5.0/docs/guide/plugin/developer_guide/java_js.html

Java-to-JavaScript Communication. Retrieved March 28, 2008, from http://java.sun.com/j2se/1.5.0/docs/guide/plugin/developer_guide/java_js.html

Jeremy M. Norman. (2005). From Gutenberg to the Internet: A Sourcebook on the History of Information Technology. Norman Publishing.

Jesse J. Garret. (2005). Ajax: A new approach to web applications. Retrieved March 28, 2008, from http://www.adaptivepath.com/ideas/essays/archives/000385.php

Joshua Duhl. (2003). White paper: Rich internet applications. Technical report, IDC. Available from http://download.macromedia.com/pub/solutions/downloads/business/idc_impact_of_rias.pdf

Kevin Hakman. (2006). Retrieved March 21, 2008, from http://www2.sys-con.com/webinararchive.cfm?registered=on&pid=wc_aw6_d1_s3_t2_hakman

LINQ: .NET Language-Integrated Query. Retrieved March 28, 2008, from http://msdn2.microsoft.com/en-us/library/bb308959.aspx

Matthew MacDonald. (2007). Pro WPF: Windows Presentation Foundation in .NET 3.0. Apress.

Michael Mahemoff. (2006). Ajax Design Patterns. O'Reilly.

Microsoft's Interoperability Principles and IE8. Retrieved March 28, 2008, from http://blogs.msdn.com/ie/archive/2008/03/03/microsoft-s-interoperability-principles-and-ie8.aspx

MonoTorrent. Retrieved March 28, 2008, from http://monotorrent.blogspot.com/2007/09/so-summer-is-finally-at-end.html

Nokia to bring Microsoft Silverlight powered experiences to millions of mobile users. Retrieved March 28, 2008, from http://www.nokia.com/A4136001?newsid=1197788

Paul E. Ceruzzi. (1998). A History of Modern Computing. The MIT Press.

Programming Ruby. Retrieved March 28, 2008, from http://www.ruby-doc.org/docs/ProgrammingRuby/html/tut_modules.html#S2

Ruby-like Mixins in Actionscript 3.0. Retrieved March 28, 2008, from http://flexonrails.net/?p=73

Scott Guthrie. (2008). Blog entry: First look at silverlight 2. Retrieved March 28, 2008, from http://weblogs.asp.net/scottgu/archive/2008/02/22/first-look-at-silverlight-2.aspx

Search enabling Silverlight and AJAX web applications. Retrieved March 28, 2008, from http://blogs.msdn.com/jhawk/archive/2007/05/23/searching-enabling-silverlight-and-ajax-web-applications.aspx

Sebastian Weber et al. (2008). Workshop on Learning Software Organizations (LSO). Rome, Italy, 2008.

Silverlight FAQ. Retrieved March 28, 2008, from http://www.microsoft.com/silverlight/overview/faq.aspx

Silverlight FAQ. Retrieved March 31, 2008, from http://www.microsoft.com/silverlight/overview/faq.aspx

Sun starts bidding adieu to mobile-specific Java. Retrieved March 28, 2008, from http://www.news.com/8301-13580_3-9800679-39.html

Sun offers JavaFX road map. Retrieved Mai 21, 2008, from http://www.infoworld.com/article/08/05/06/Sun-offers-JavaFX-road-map_1.html

Sun's JavaFX tools to interop with Adobe. Retrieved March 28, 2008, from http://www.regdeveloper.co.uk/2008/01/24/javafx_tools_adobe/

System Requirements for JRE 6.0. Retrieved March 28, 2008, from http://java.com/en/download/help/6000011000.xml

Tamarin Project. Retrieved March 28, 2008, from http://www.mozilla.org/projects/tamarin/

The Counter.com. Retrieved March 27, 2008, from http://www.thecounter.com/stats/2008/February/browser.php

The dom and javascript. Retrieved March 26, 2008, from http://developer.mozilla.org/en/docs/The_DOM_and_JavaScript

The DOM and JavaScript. Retrieved March 27, 2008, from http://developer.mozilla.org/en/docs/The_DOM_and_JavaScript

The JavaFX Script Programming Language. Retrived April 9, 2008, from https://openjfx.dev.java.net/JavaFX_Programming_Language.html

The XMLHttpRequest Object. Retrieved March 27, 2008, from http://www.w3.org/TR/XMLHttpRequest/

Thermo. Retrieved March 28, 2008, from http://labs.adobe.com/wiki/index.php/Thermo

Tim O'Reilly. (2005). What is web 2.0. Retrieved March 27, 2008, from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

TR10: Offline Web Applications. Retrieved March 30, 2008, from http://www.technologyreview.com/read_article.aspx?ch=specialsections&sc=emerging08&id=20245

WHATWG FAQ. Retrieved March 24, 2008, from http://wiki.whatwg.org/index.php?title=FAQ&oldid=2907#What_are_.E2.80.9CWeb_Applications.E2.80.9D.3F

KEY TERMS & DEFINITIONS

Web application: A Web application is an application accessed over the WWW using a Web browser. It is built on Web standards. Additionally, proprietary Web technologies may be used.

Rich Internet Application: Applications called RIAs provide a more intuitive, responsive, and effective user experience. This is done by utilizing user interface components and behaviors know from desktop applications.

Click-Wait-and-Refresh-Cycle: This term was coined by Kevin Hakman (2006). It describes the way users interact with traditional Web applications. A user clicks on a button or link, and the request is sent to the server and processed. The user waits until the results are returned to the Web browser, which refreshes the presentation.

Rich User Experience: The experience of a user using traditional Web applications and websites is characterized by the Click-Wait-and-Refresh-Cycle and the available set of user interface components. Thus, a rich experience is built up by adding additional interface components and behaviors and the Click-Wait-and-Refresh-Cycle is avoided by retrieving and presenting data from the server without refreshing the whole page.

All-audience applications: Applications potentially targeting every Internet user. Thus, this type of application has to take care that each potential user can access and use the application, regardless of which browser is installed on his system. If browser plug-ins are needed, only plug-ins with a very high market penetration are sufficient.

AJAX: AJAX stands for Asynchronous JavaScript + XML. It is not one technology, but a combination of technologies. These include HTML, Cascading Stylesheets (CSS), Document Object Model (DOM), XML, Extensible Stylesheet Language Transformations (XSLT), XMLHttpRequest, and JavaScript. HTML and CSS are used for presentation. DOM allows manipulation of the presentation and interaction. XML and XSLT are used for data interchange and manipulation. XMLHttpRequest allows retrieval of data. JavaScript is used to define the underlying logic and interaction of the other technologies.

ENDNOTES

[1] http://lobobrowser.org/

2 http://www.curl.com/

3 http://www.omnis.net/index.html? detail=overview

4 http://labs.mozilla.com/2007/10/prism/

5 http://www.aptana.com

6 Widget is the term commonly used to describe user interface components in the context of a JavaScript framework.

7 Browser bugs are a source of problems with HTML and JavaScript development: http://www.positioniseverything.net

8 http://www.prototypejs.org

9 Meaning: Windows, Mac OS X, and the common Linux distributions.

10 http://jquery.com

11 http://redesign.dojotoolkit.org/jsdoc/dojo/HEAD/dojo.behavior

12 IE 6 has problems with PNG transparency.

13 http://www.mono-project.com

14 http://www.mono-project.com/Moonlight

15 The CLR is the virtual machine of Microsoft's .NET framework. It is Microsoft's implementation of the Common Language Infrastructure.

16 Pronounced 'Zammel'.

17 Code that is running inside the CLR is called managed code.

18 Play a video while it is still downloading.

19 In combination with the Windows Media services platform

20 See source code of http://www.microsoft.com/silverlight/

21 Flex uses Cascading Style Sheets (CSS) to style the appearance of components.

22 Mac OS X is not on the list for the SDK, but on the list for Flex Builder 3, and Flex Builder 3 comes with the SDK.

23 Compilers only.

24 Flex allows managing states, enabling back-button handling in a way the user would expect from a browser.

25 Allow accessing HTTP and SOAP services.

26 This is not allowed by the compiler in strict mode.

27 But there exists an easy workaround using JavaScript.

28 US, Canada, UK, Germany, France, Japan.

29 China, S. Korea, Russia, India and Taiwan.

30 SVG is not supported by the Flash player, but by the Flex compiler.

31 Link to the Java Web Start file of the Pad: http://download.java.net/general/openjfx/demos/javafxpad.jnlp

32 JavaFX preview in NetBeans: http://wiki.netbeans.org/JavaFXPluginPreviewSpecification

33 For Eclipse, an additional plug-in is needed.

34 http://lobobrowser.org/cobra.jsp

35 http://commons.apache.org/

36 http://java.sun.com/javase/technologies/desktop/java3d/

37 An operation is the JavaFX counterpart of Java methods. JavaFX is used to distinguish between operations and functions. In the latest version of the language, there are only functions.

38 US, Canada, UK, France, Germany, Japan

39 http://www.javafx.com

40 http://www.bubblemark.com/

41 http://java.sun.com/javafx/tutorials/jfx_nb_getting_started/