

RIA 플랫폼 개발환경 비교

지정훈^{○*}, 박규태*, 우균*, 김원영**, 최완**

^{*}부산대학교 컴퓨터공학과

^{**}한국전자통신연구원

e-mail:jhji@pusan.ac.kr

Comparing Development Environment for RIA Platforms

Jeong-Hoon Ji^{○*}, Gyu-Tae Park*, Gyun Woo*, Won-Young Kim**, Wan Choi**

^{*}Dept of Computer Engineering, Pusan National University

^{**}Digital Home Research Division, Electronics and Telecommunications Research Institute

요 약

인터넷의 확산으로 소프트웨어 산업구조는 패키지 방식에서 웹을 중심으로 한 방식으로 변화되었다. 소프트웨어 배포 및 사용이 웹을 통해 이루어지고 사용자들도 자신이 원하는 콘텐츠를 직접 제작할 수 있게 되었다. 이와 더불어, 웹 기반 소프트웨어에 대한 사용자 요구사항도 더욱 많아졌다. 사용자들은 멀티미디어 지원, 상호작용, 데스크톱 수준의 시각적으로 화려한 인터페이스(UI)를 웹 브라우저를 통해 사용하기를 원한다. 하지만, HTML과 스크립트 언어 기반의 웹 시스템만으로는 사용자 요구사항을 만족시키기가 어렵다. RIA(Rich Internet Application)는 HTML의 한계를 극복하고, 사용자 요구사항을 충족시키기 위해 제안된 모델이다. 향후 RIA 기술은 많은 분야의 소프트웨어 개발에 사용될 것이다. 본 논문에서는 RIA 기술에 대해 알아보고 RIA 개발을 위해 필요한 요소들을 정의하고 RIA 개발환경들의 장단점을 비교 평가한다. 또한 각 플랫폼의 성공적인 응용사례를 살펴봄으로써 RIA 개발환경 선택에 도움을 주고자 한다.

1. 서 론

지금까지의 대부분 웹 시스템들은 HTML을 기반으로 PHP, ASP, JSP와 같은 프로그래밍 언어를 이용해 제작된 인터페이스를 제공하였다. HTML 기반의 웹 시스템들은 구조가 간단하여 제작하기 쉽다는 장점이 있지만, 멀티미디어나 사용자와의 상호작용, 데스크톱 환경과 같은 화려한 사용자 인터페이스를 제공하기 어렵기 때문에 HTML 만으로는 사용자들의 요구사항을 만족시키기에 한계가 있다. 이 때문에 최근에는 HTML의 한계를 보완하고 사용자 참여와 요구사항들을 수용할 수 있는 리치 인터넷 어플리케이션(RIA : Rich Internet Application)이 등장하였다[1].

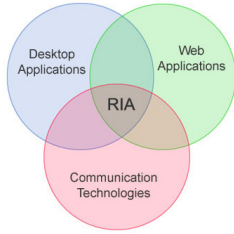
RIA 기술은 웹 브라우저를 통해 편리한 사용자 인터페이스를 제공하고 HTML의 한계를 극복하여 사용자 중심의 풍부한 GUI를 제공한다. 또한, RIA는 사용자 참여를 통한 웹 콘텐츠(contents) 개발을 위한 개발환경을 제공한다. 다양한 콘텐츠 제공을 위해서는 사용자 참여가 필수적이다[2]. 콘텐츠 공급자뿐만 아니라 사용자가 직접 콘텐츠를 개발해 배포하면 지금보다 더욱 다양한 콘텐츠를 개발할 수 있다. 이를 위해서는 무엇보다도 개발이 쉬워야 한다. 소프트웨어나 콘텐츠 개발 경험이 없는 사용자도 자

신이 원하는 기능을 구현할 수 있을 정도로 직관적이고 쉬워야 한다.

본 논문에서는 RIA 환경에 대해서 알아보고, RIA 개발을 위해 필요한 요소들을 정의한다. 그리고 RIA 플랫폼들의 특징을 분석하고 개발환경의 장단점을 비교 분석한다. RIA 개발환경으로는 Ajax, 플렉스(Flex), 실버라이트(Silverlight)[3], 오픈라즐로(Openlazo)[4]를 비교대상으로 선택하였다. 본 논문에서는 웹 어플리케이션 및 콘텐츠 개발을 위해 필요한 구성요소 및 요구사항들을 선정하고, 이를 기준으로 각 플랫폼의 개발도구들을 평가한다. 또한 각 플랫폼의 성공적인 응용 사례를 살펴봄으로써 RIA 기반의 웹 응용 소프트웨어 개발자들의 개발도구 선택에 도움을 주고자한다.

2. RIA 플랫폼

RIA는 과거의 정적인 웹페이지에서 벗어나 동적으로 서비스 제공이 가능하고, 데스크톱 수준의 화려한 인터페이스를 통해 사용자와의 상호작용이 가능하다[5]. RIA는 웹 브라우저를 클라이언트로 사용하며 사용자는 웹 브라우저와의 상호작용을 통해 서비스를 제공받는다. (그림 1)는 RIA 모델을 나타낸다.



(그림 1) RIA(Rich Internet Application) 모델

2.1 Ajax

Ajax는 “Asynchronous JavaScript + XML”로 자바스크립트와 XML을 활용한 RIA 플랫폼이다. Ajax에서는 클라이언트와 웹 브라우저 사이에 XML을 이용해 데이터를 교환한다. Ajax의 장점은 서버와의 데이터 전송에 있어, 이벤트 처리에 필요한 데이터만 전송받아 사용자의 대화식 요청을 효율적으로 처리할 수 있다는 것이다. 또한 Ajax는 비동기 통신을 하므로 데이터 전송을 하고 나서도 사용자는 다른 작업을 할 수 있다.

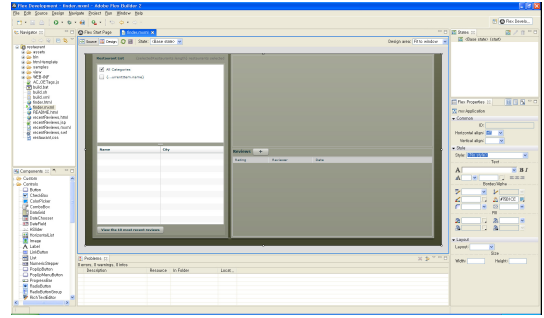
Ajax 어플리케이션 개발에는 XHTML, JavaScript, CSS, XML Http Request (XHR) 등의 기술들이 이용된다. 또한 PHP, Python, Ruby와 같은 다른 스크립트 언어와의 연동이 가능하다. 그리고 Ajax용 어플리케이션 개발도구로는 오픈소스이며, 무료로 사용할 수 있는 이클립스(eclipse) 기반의 도구들이 사용되고 있다.

2.2 Flex

Adobe사에서 개발한 플래시(Flex)는 플래시(Flash) 기반의 RIA 플랫폼이다. 플래시는 플래시 기술을 바탕으로 하여 애니메이션과 사용자와의 상호작용이 뛰어난 UI를 제공한다. 플래시는 플래시 플레이어(Flash Player)를 런타임 엔진으로 사용한다. 플래시용 콘텐츠들은 플래시 플레이어를 이용해 독립적으로 수행되거나, 웹 브라우저를 통해 수행된다. 플래시 플레이어는 인터넷 사용자들의 90%이상 사용하고 있을 정도로 범용성이 뛰어나다.

플렉스 어플리케이션 개발은 데스크톱용 어플리케이션 개발과정과 비슷하다. 플렉스는 개발환경으로 Flex Builder를 제공한다. Flex Builder는 플래시를 이용한 UI 디자인과 이벤트 처리를 위한 코딩을 할 수 있다. 플렉스는 개발언어로 MXML이라는 마크업 언어와 ActionScript 사용한다. Flex Builder를 이용해 개발된 어플리케이션은 swf형식으로 만들어져

배포할 수 있다. swf 파일은 플래시 플레이어에 의해 수행되거나 플래시 플레이어가 플러그인으로 포함된 웹 브라우저를 통해 수행된다. 그림 4는 Flex Builder를 이용한 개발 환경을 보여준다.

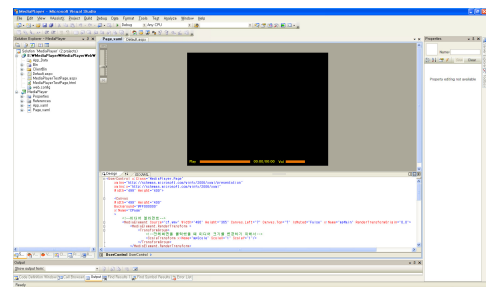


(그림 2) Flex Builder 개발환경

2.3 Silverlight

실버라이트(silverlight)는 마이크로소프트에서 개발한 차세대 RIA 플랫폼으로 2007년도 9월에 소개되었다. 실버라이트의 구조는 Flex와 유사하다. 그리고 실버라이트는 미디어 기능 지원이 우수하다. 실버라이트는 Flex와 동일하게 실버라이트 런타임 엔진을 설치해야 사용할 수 있다.

마이크로소프트에서는 실버라이트 개발환경으로 익스프레션 스튜디오(MS Expression Studio)를 개발하였다. 익스프레션 스튜디오는 Web, Design, Blend, Media의 네 가지 패키지로 구성되어 있으며 RIA용 콘텐츠 개발을 위한 환경을 제공한다. 그림 6은 실버라이트 개발환경과 XAML을 이용한 UI 디자인을 보여준다. 실버라이트의 UI 디자인은 XAML이라는 XML 기반의 언어를 이용하고, 이벤트 처리는 DLR(Dynamic Language Runtime)을 통해 .Net 개발환경에서 지원하는 C++, C#, JScript 뿐만 아니



(그림 3) 실버라이트 개발환경

라 Python, Ruby 등의 스크립트 언어를 사용할 수

있다. 이들 언어로 작성되어진 코드는 CLR 엔진에 의해 수행 가능한 중간 코드로 컴파일 된다.

2.4 OpenLazlo

오픈라즐로(OpenLazlo:라즐로)는 공개 소스 프로젝트로 개발되는 RIA 플랫폼이다. 오픈라즐로는 라즐로 서버와 클라이언트(웹 브라우저)로 구성된다. 라즐로 서버는 JavaEE 서블릿 컨테이너에서 동작하는 자바 어플리케이션으로 LZX로 작성되어진 소스 코드를 컴파일하여 바이트코드를 생성하여 웹 브라우저로 보낸다. 라즐로는 런타임 실행기로 플래시 플레이어를 사용한다. 이 때문에 라즐로 서버에서는 swf 형식의 바이트코드를 생성한다.

라즐로 클라이언트는 LZX 프로그래밍 언어로 작성된다. LZX는 XML 태그와 JavaScript API로 구성된다. 오픈라즐로의 UI인터페이스는 XML 태그를 이용해 작성되고, 이벤트 처리 코드는 JavaScript API를 이용해 작성한다. 그리고 컴파일러는 LZX 파일과 미디어 셋(이미지, 사운드 등)을 합쳐서 swf 바이트코드를 생성한다.

3. 비교 기준

본 절에서는 RIA를 저작하기 위해 개발환경이 갖추어야 하는 기준들을 제시한다. 이들 기준은 비록 주관적인 관점에 의해 정의하였지만, RIA 개발을 위해서는 필수적으로 포함해야 하는 기능들이다. 다음 절에서는 RIA 개발 시 개발환경 선택에 도움을 주고자 이들 기준을 중심으로 각 도구를 비교하였다.

3.1. 통합개발 환경

통합개발환경은 디자인 도구, 에디터, 컴파일러, 디버거 등 어플리케이션 개발에 꼭 필요한 구성요소이다. 통합개발환경 지원여부는 개발비용에도 직접적인 영향을 미친다. 어플리케이션 개발에 필요한 구성요소들을 제공하는 통합개발환경을 지원하는가?

3.2. GUI

GUI 제작 도구를 지원하는가? GUI를 위한 언어를 지원하는가?

3.3. 컴포넌트 지원

다양한 컴포넌트를 지원하는가? 컴포넌트 재사용이 가능한가?

3.4. 지원 언어

다양한 프로그래밍 언어를 지원하고 있는가? 사용자가 쉽게 언어를 익혀서 어플리케이션을 개발할 수 있는가?

3.5. 미디어 지원

텍스트, 이미지, 소리, 동영상 처리를 위한 미디어 처리가 쉬운가?

3.6. 디버깅

어플리케이션 디버깅이 쉬운가? 디버깅 환경이 지원되는가?

3.7. 배포

콘텐츠의 배포가 용이한가?

3.8. 모바일 환경 지원

PDA와 같은 모바일 서비스 개발을 지원하는가?

3.9. 응답성

사용자의 요청에 대해 원활한 응답을 제공하는가?

3.10. 개발 용이성

일반 사용자들도 콘텐츠 개발에 참여 가능할 정도로 개발환경 사용이 쉬운가?

4. RIA 개발환경 비교

본 절에서는 Ajax, 플렉스, 실버라이트, 오픈라즐로의 RIA 개발환경들에 대해 3절에서 제시한 기준에 따라 상대적 만족도를 점수로 환산하였다. 표1은 각 RIA 개발환경에 대한 상대적 만족도를 보여준다.

<표 1> RIA 저작 도구의 비교 결과

비교기준	RIA 플랫폼			
	Ajax	플렉스	실버라이트	오픈라즐로
통합개발	8	10	10	6
GUI	7	10	9	8
컴포넌트	9	10	9	9
지원언어	8	8	10	7
미디어	8	10	10	7
디버깅	8	8	10	8
배포	10	10	10	10
모바일	8	10	8	7
응답성	10	10	10	8
용이성	10	10	10	9
총점	86	96	96	79

표1의 비교기준은 3절에서 제시한 각 비교 기준을 나타낸다. 그리고 RIA 플랫폼은 네 가지 RIA 플랫폼을 나타내고, 2행부터는 각 항목에 대한 점수를 나타낸다. 끝으로 테이블 마지막의 총점 행은 각 기준의 점수를 모두 합한 값이다.

먼저, 통합개발환경은 Ajax와 플렉스는 이클립스 기반의 통합개발환경을 제공한다. Ajax의 경우, 이클립스 플러그인이 공개소스로 개발되어 배포되고 있고, 플렉스의 경우 Adobe에서 상용으로 Flex Builder를 제공하고 있다. 실버라이트의 경우, .NET 개발환경을 사용하고 있으며, 오픈라즐로는 공개소스로 IDE4Lazlo를 플러그인형태로 제공하고 있다. 플렉스와 실버라이트는 UI 디자이너, 에디터, 디버거 등 우수한 성능의 통합개발환경이 제공되고 있어 높은 점수를 주었고, 이에 반해, Ajax와 오픈라즐로는 공개소스지만 성능이 미치지 못해 상대적으로 낮은 점수를 주었다.

GUI 지원의 경우에도 플렉스와 실버라이트가 뛰어나다. 플렉스의 경우, 플래시를 이용해 동적이고 화려한 UI제작이 가능하고 실버라이트의 경우, 익스프레션 스튜디오를 이용해 데스크톱 수준의 UI 제작이 가능하다. 또한 이들은 XML 형식의 GUI 전용 마크업 언어를 제공하고 있다.

컴포넌트의 경우, 네 가지 플랫폼 모두 잘 정의되어 있으며, 재사용성 또한 우수하다. 지원언어와 미디어 지원 부분에서는 실버라이트가 우수하다. 실버라이트는 .NET 프레임워크의 CLR 런타임을 통해 C++, C#, JScript, Python, Ruby 등의 다양한 언어를 지원하고 있다. 미디어 지원 부분은 플렉스와 실버라이트가 비슷하지만 실버라이트가 VC-1 코덱을 통한 HD급의 스트리밍 서비스를 쉽게 할 수 있다는 점에서 더욱 우수하다.

콘텐츠 배포는 RIA 플랫폼 모두 별도의 설치과정 없이 배포할 수 있어서 우수하다고 할 수 있다. 모바일로의 확장은 플렉스의 경우, 플래시 플레이어만 있으면 별도의 변환 없이 확장할 수 있어 우수한 점수를 얻었다. 응답성과 용이성의 경우, Ajax와 플렉스, 실버라이트가 오픈라즐로에 비해 우수하다고 할 수 있다. 응답성은 GUI와 다양한 컴포넌트를 통해 사용자 상호작용을 효율적으로 처리할 수 있으며, 용이성은 GUI 디자인과 코딩을 분리시킴으로써 분업이 가능하고, 일반 사용자들이 약간의 교육을 통해 간단한 콘텐츠를 제작할 수 있을 정도로 개발환경의 지원이 우수하다.

마지막으로 각 항목들의 총점을 계산해 보았다. 비교 결과 총점의 합은 플렉스와 실버라이트가 가장 높은 점수를 얻었으며, 다음으로 Ajax와 오픈라즐로의 순서이다. RIA 플랫폼의 전체적인 비교결과 Adobe의 플렉스와 마이크로소프트의 실버라이트가 통합개발환경, GUI, 미디어 지원 등 많은 분야에서 우수하다는 결론을 얻을 수 있었다.

5. 결론

본 논문에서는 차세대 인터넷 플랫폼인 RIA 환경에 대해 살펴보고, RIA 플랫폼의 개발환경을 비교분석하였다. 본 논문에서는 이들 네 가지 플랫폼과 개발환경에 대해 10가지 기준에 대한 평가를 하였다. 비교결과를 요약하면 다음과 같다.

- Ajax - 사용자 응답성과 개발 용이성이 우수.
- 플렉스 - 상호작용 가능한 GUI 제작이 우수.
- 실버라이트 - 기존의 .NET 개발자들이 쉽게 사용할 수 있으며, 미디어 지원이 우수
- 오픈라즐로 - 공개 소스로 XML 기반의 RIA 콘텐츠를 개발할 수 있다.

RIA 플랫폼들의 특징과 개발환경을 살펴본 결과, 상호작용과 데스크톱 수준의 UI 지원, 고속 미디어 처리와 같은 요구사항들에 대해 플렉스와 실버라이트가 잘 지원하고 있다. 플렉스와 실버라이트는 XML 기반의 언어를 이용해 UI 디자인과 코드를 분리하고 있으며, 우수한 기능의 통합개발환경을 지원하고 있다. 또한 고속의 실행엔진을 바탕으로 별도의 설치 없이 콘텐츠를 배포할 수 있어 콘텐츠 이용의 편의성이 높다. 향후에는 플렉스와 실버라이트를 이용해 데스크톱 수준의 소프트웨어를 RIA 개발환경을 이용해 구현해 보고 소프트웨어 분야별로 알맞은 RIA 플랫폼에 대해 분석할 것이다.

참고문헌

- [1] 강안나, 고대식, "Ajax 적용사례 분석", 한국정보기술학회 2007 하계학술대회, pp. 458-463, 2007.
- [2] 조대제, 김시관, "웹 2.0을 위한 리치인터넷 어플리케이션 개발 도구에 관한 연구", 한국정보기술학회논문지, Vol.6(4):pp. 115-127, 2008.
- [3] Silverlight, <http://www.microsoft.com/silverlight>
- [4] OpenLazlo, <http://www.openlazlo.org/>
- [5] 송희정, 백종현, "RIA 구현 기술 전략에 관한 연구", 제32회 한국정보과학회 추계학술발표 논문집, Vol.32(2):pp. 967-969, 2005.