RIA 구현 기술 전략에 관한 연구

송희정, 백종현
대우정보시스템 정보기술연구팀
hjsong@disc.co.kr, baegjh@disc.co.kr

A Study on Implemention Strategy of RIA

Heejung Song, Jonghyun Baeg
Information Technology Team, Daewoo Information Systems Co., Ltd.

요 약

이제 웹 어플리케이션도 과거의 정적이고 단방향의 형태에서 벗어나 다양한 화면과 인터랙티브한 어플리케이션으로의 전환의 시점을 맞이하였다. 이른바 RIA(Rich Internet Application)이라 불리는 새로운 형태의 어플리케이션 구현에 있어서 어떠한 기술을 이용하여 구현을 할 것인지, 어떠한 솔루션을 채택해야 하는지에 관한 것이 새로운 이슈로 부각되고 있다. 본 논문에서는 이에 대한 방안으로 웹 프레임웍, 웹 표준, 솔루션의 세가지 형태를 살펴보고, 상황에 맞는 기술전략 가이드를 제시한다.

1. 서 론

최근 웹 어플리케이션 개발의 패러다임이 바뀌고 있다. 과거 정적인 페이지에서 벗어나 네트웍상의 각종 서비스를 제공하고, C/S 로만 가능하던 기능을 구현하기 위한 연구와 기술의 도입이 활발하게 이루어지고 있다. 하지만, 이러한 기능을 위해선 기존의 기술로는 불가능한 부분이거나, 구현에 있어서는 그 생산성이나 구현후 효과에 취약한 측면이 있다 특히 구현 후 사후 관리에 있어서도 효율적이지 못하다. 본 논문에서는 이에 대한 방안을 살펴보고 각각의 특징과 어떤 상황에서 적합한지에 대한 가이드를 제시할 것이다.

2. 웹 어플리케이션의 개발 동향

웹이란 개념이 대중화된지 어느덧 10 년 여의 시간이 흘렀다. 이제 웹이란 것은 단순한 컨텐츠의 보여주기식 기능에서 벗어나 기존의 레거시 시스템을 대체하고 있다. 나아가서 클라이언트는 과거 C/S 와 같이 혹은 그 이상의 동적이고, 자동화된, 사용자 중심의 인터페이스를 요구하고 있으며 이를 일컬어 RIA(Rich Internet Applications)라고 부르고 있다.

RIA 는 다음과 같은 요구조건을 만족해야 한다.

- ✓ 동적인 사용자 화면 : 전체적인 화면의 갱신 없이 부분적인 화면정보 갱신이나, 동적인 화면 혹은 다양한 미디어가 embedding 된 통합된 화면 처리
- ✓ RAD : 개발자는 단지 전체적인 화면의 흐름만을 제어한다든지, 단순화된 개발화면, 코드작업의 최소화
- ✓ 유비쿼터스 환경 : 클라이언트의 환경에 구애받지 않고 다양한 기술표준과 호환가능, 복잡한 설치절차없이 구동가능

RIA 를 구현하고 다양한 사용자의 요구사항에 부응하기 위한 솔루션들이 등장하고 있고, 많은 기술을 요하고 있으나, 이에 대한 적절한 가이드나, 상황에 따른 최적의 방안을 모색하기란 쉽지가 않은 것이 현실이다. 특히 개발자 측면에서는 Presentation tier 의 구현이야말로 사용자의 요구사항에 따라 그 형태나 구현 기술이 판이하게 달라지므로 많은 어려움과 혼란을 겪게 된다.

개발에서 가장 먼저 고려하게 되는 것중 하나가 어떤 기술을 이용하여 구현할 것인지에 대한 선택이 될 것이다. Presentation tier 에서는 크게 세가지 방향을 고려할 수 있다.

먼저 가장 일반적으로 쓸 수 있는 Web Framework 을 사용하는 방법으로 많이 알려진 Struts 나 Webwork, Spring MVC 를 사용하여 구현하게 되는 것이다.

그 다음으로는 사용범위가 확장되고 있는 X-Internet 을 주창하는 솔루션들을 들 수 있다. 이러한 솔루션들은 ActiveX 나 Applet 등을 이용하여 웹 화면을 C/S 에서의 화면과 유사한 형태로 구현할 수 있게끔 도와준다. 마지막으로는 기술 표준을 따르는 것으로써, 아직은 많이 사용되고 있지는 않지만, JSF 나 AJAX 를 이용하여 동적인 화면을 구현하는 방법이 있을 수 있다. 이제 이 세가지 방안을 하나씩 살펴보면서 각 방안이 어떤 상황에서 적합한지, 장단점을 살펴보도록 할 것이다.

3. 오픈 소스 웹 프레임웍

Jakarta 의 Struts 나 OpenSympony 의 Webwork 와 같은 Presentation tier 의 프레임웍은 이미 일반적으로 많이 보급되어 실질적인 표준으로 취급되면서 많은 어플리케이션에서 사용되고 있는 검증된 프레임웍이다.

이들은 프레임웍의 소스를 공개하여 그 특성에 맞추어 개발자들이 개발을 할 수 있으며, 현재 나와있는 대부분의 JAVA IDE 에서는 Struts 기반의 어플리케이션을 자동으로 생성할 수 있는 Wizard 기능을 제공하고 있다.

프레임웍은 CBD 방법론이 대두되면서 그 필요성과 사용이 확대되었으며 프레임웍은 컴포넌트가 구동되는 환경을 제공할 뿐만 아니라, 자체적인 UI 컴포넌트를 제공해 주기도 한다. 그런면에서 RIA 를 구현하기 용이하며, 이미 구현된 많은 프레임웍 기반의 어플리케이션들이 그 프레임웍의 기능의 검증과 RIA 구현의 모델이 될 수 있다.

<표 1> 오픈소스 웹 프레임웍 비교

프레임웍	특징
Struts	장점 • 대표적인 MVC 기반 오픈소스 프레임웍 • 장점 JSF 나 AJAX 와의 연동 적합 • Tiles 를 통한 다양한 Layout 지원 • Client- and server-side validation 단점 • 테스팅이 가능하나, 제한적임 • ActionForm 으로 인한 구현의 어려움
Webwork	장점 • 유연하고, 직관적이므로 개발이 용이 • Taglib 의 커스터마이징이 용이하여 다양한 화면구성 가능 • OGNL 사용 단점 • 참조문서와 샘플이가 적다 • Client-side validation 이 미흡
Spring MVC	장점 • IoC 개념으로 테스팅이 용이함 • HTML, Tiles,Excel,PDF 다양한 화면처리가능 단점 • 아직은 실제 사용은 적은 상태 • JSP 코딩 분량이 많음 • 지나친 유연성으로 common 한 컨트롤러 부족
Tapestry	장점 • 템플릿을 HTML로 관리하여 화면 개발 및 관리가 용이하고, 포틀릿을 지원함 • 단순하며 확장이 쉽다. 단점 • 문서가 빈약하고 아직은 초기단계임 • 테스팅이 어려움

프레임웍 기반의 개발이나 유지보수는 기존의 Model1 방식보다는 훨씬 체계적이나 프레임웍에 맞추어 정해진 형태대로 구현을 해야하며, Taglib 나 프레임웍의 이해없이는 구현이 어렵다. 실제 프로젝트에 적용시 기존의 생산성과 같아지기 위해선 2 개월 정도의 훈련/개발 기간이 필요하며, 초기에는 1.5 배 정도의 시간이 더 소요되므로 이를 감안해야 한다. 또한 RIA 는 브라우저만을 통해 접근하는 초기 웹 어플리케이션에서 벗어나 다양한 유비쿼터스 환경을 고려해야 한다. 그러므로 구축할 RIA 의 요구사항에서 클라이언트의 환경과 접근형태가 다양할 경우 추가의 모듈이 필요하게 된다.

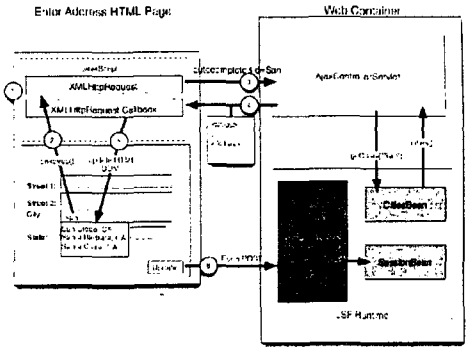
4. 표준 스펙

JSF(Java Server Faces)는 웹 어플리케이션의 UI 컴포넌트를 위한 표준 프레임웍 구현을 목적으로 만들어진 표준 스펙이다. JSF 에서 제공하는 기능으로는

- UI 컴포넌트 모델/이벤트 처리 모델/ 렌더링 모델
- Validation 프레임웍

등 이 있다. JSF 는 기존 웹 어플리케이션에서 취약했던 화면의 처리를 위한 것으로, 오라클이나 IBM, 오픈소스 계열에서 이를 준수하는 다양한 툴이나 제품이 속속 등장하고 있는 추세이다.

이 표준에 덧붙여 AJAX (Asynchronous Javascript And XML) 라는 아키텍쳐를 함께 응용하여 동적인 RIA를 구현할 수 있다.



(그림 1) AJAX+JSF

AJAX는 자바스크립트와 XMLHttpRequest를 이용하여 클라이언트가 인식하지 못하지만, 실제로 어떤 이벤트가 발생할 때마다 서버와의 인터랙티브하게 동작하며 현재는 초기단계로서 실시간 데이터 검증이나, 문장 자동완성, 데이터 갱신등에 걸쳐 다양한 형태로 적용되고 있는 형편이다.

AJAX 와 JSF 는 공통적으로 XML 데이터의 처리에 강하고 유연하다는 특징을 가지고 있다. 그러므로 웹서비스 기술이나 XML 을 사용하여 구현하는 어플리케이션에 유리하고, 기존의 취약했던 다이나믹한 웹 화면의 구성을 가능하게 할 수 있다. 각종 IDE 에서 JSF 를 지원하고 있고, 관련 컴포넌트들이 다양하게 등장하고 있으며, AJAX 는 기존의 XML 과 자바스크립트로 구현하므로 접근도 쉽다는 장점이 있는 반면, AJAX 의 근간은 자바스크립트이므로 다음과 같은 단점을 고려해야 한다.

- 브라우저에 종속적-제품, 버전별로 호환 확인이 필요하다.
- 디버깅이나 테스팅의 어렵다.
- 화면에서 소스보기를 통해 코드가 드러난다.

따라서 이러한 부분에 민감한 모듈 부분은 JSF 로, 간단하며 이벤트성 모듈은 AJAX 로 적절히 혼합하여 구현하는 방안을 추천한다. 또 JSF 는 MVC 에서 View 에 집중되어 있으므로 기존에 Struts 와 같은 프레임웍을 사용하는 경우, 화면의 렌더링 만을 교체하여 JSF 의 적용도 무리없이 가능하다.

5. X-Internet Solutions

RIA 를 구현하기 위한 가장 대표적인 솔루션으로 등장하는 것이 바로 X-Intenet 솔루션이다. 실시간의 양방향 통신과 다양한 디바이스의 지원, C/S 처럼 처리 가능한 화면등으로, eXtended /eXecutable Internet 을 일컬어 X-Internet 이라 부른다.

2000 년 포레스터 리서치에서 처음 언급된 X-Internet 은 브라우저에 국한되지 않고 다양한 디바이스를 지원하는 어플리케이션을 대상으로 하면서 데이터 렌더링이 탁월한 기능으로 C/S 와 같은 화면을 구성하는데 용이하다.

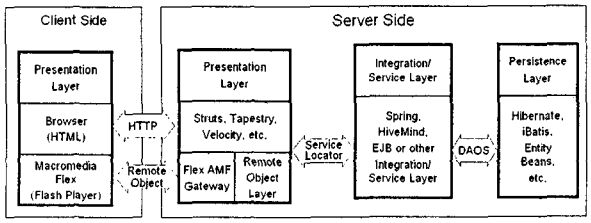
국내에도 X-Internet 을 이용한 프로젝트가 점차 늘어나고 있으며, 다양한 솔루션들이 등장한 상태이다.

대표적인 툴로는 Macromedia 의 Flex 나 오픈 소스계열의 Lazlo, Applet 기반의 AltioLive, 그 외 ActiveX 기반의 TrustForm, Miflatform 등이 있으며, 이러한 툴들은 화면 작성을 위한 WISWIG 디자인 화면을 제공한다. X-Internet 의 동작원리는 서버에서 클라이언트로 데이터만을 업데이트하고, 사용자 인터페이스는 클라이언트 시스템에서 다시 그려주는 구조로서 인터넷의 HTTP 트래픽을 줄여 성능을 향상시키는 아키텍처이다.

이러한 솔루션을 쓰는 경우, C/S 에 익숙한 사용자들이 많은 경우, 웹화면에서 C/S 와 가장 가깝게 구현이 가능하므로 그 만족감이 높고, 개발자들도 특별한 코딩기술을 많이 요하지 않아서 개발 생산성도 그만큼 높게 된다. 또 반드시 웹 브라우저가 아니라 할지라도 구현하기에 따라서 모바일용 어플리케이션이 될 수도 있고, 일반 데스크탑에서 구동되는 어플리케이션처럼 사용도 가능하다.

그러나, 화면을 렌더링 하는 기술이나, 화면 작성시에 툴에 종속되어 개발하게 되므로, 해당 솔루션이 기술지원을 하지 않는다든지, 다른 기술이나 환경으로 교체시에 솔루션에서 지원하는 제품이 아닌 경우, 구현이 불가능하는 등 그만큼 제약이 뒤따르게 된다.

또 개발자의 스킬업 측면에서 특정 솔루션에만 종속되어 단순한 화면 작성자로 전락해 버릴 우려가 있다. 통합의 측면에서도 만약 기존의 Struts 와 같은 웹프레임웍을 사용하고 있다면, 이것과 함께 병행하여 사용하되, 솔루션 자체에도 MVC 프레임웍의 기능을 하는 영역이 있으므로 이에 대한 고려가 선행되어야 한다.



(그림 2) Flex 와 웹 프레임웍의 통합

6. 결 론

앞에서 살펴본 바와 같이 상황에 따라 다양한 형태로 RIA 를 구축할 수 있다. 하지만 각각의 특성과 장단점이 있으므로 이를 고려하여 어떤 기술을 사용할 것인지에 대해서는 면밀히 고려할 필요가 있다.

그리고, 어떤 하나의 기술만을 고집하지 않고, 여러 기술을 혼합하여 구현하는 것도 좋은 대안이 될 수 있다. 몇 가지 케이스 별로 나누어 살펴보면 다음과 같다.

- 다양한 Device 를 지원해야 한다.

구현하는 어플리케이션이 PC, PDA, Mobile 을 통해 접근하는 것이면, X-Internet 솔루션을 통해 구현하는 방법이 적절하다.

- 개발자의 자바 숙련도가 높다.

개발자들이 이미 웹 개발을 다수 해본 경험이 있다면, 웹 프레임웍을 이용하여 구현하고, 복잡한 UI 나 기존의 웹 프레임웍과 JSP 로 불가능한 구현인 경우에는 JSF 나 AJAX 를 함께 혼용하여 구현한다.

- 기존의 프레임웍이 존재한다.

기존의 프레임웍이 있는 경우, 간단하게 화면만을 교체할 경우에는 AJAX 와 JSF 를 통해 구현하는 방법을 우선적으로 고려하고, 그 기능으로 부족하다면, X-Internet 솔루션을 도입하여 구현하도록 한다.

- 사용자들이 C/S 의 화면을 고집한다.

웹으로 C/S 와 똑 같은 화면을 구현하기는 어려운 일이다. 만약 개발자와 사용자가 모두 C/S 에 익숙하다면, X-Internet 솔루션을 도입하는 것을 검토한다. 주의할 점은 솔루션을 도입한다 하더라도 기본적인 이벤트에 대한 코딩은 수반되어야 한다.

- 특정 벤더에 독립적이어야 한다.

차후 다른 제품으로의 이전이나, 전략적으로 특정 제품군에 독립적으로 가고자 한다면, 철저하게 표준을 따르는 것을 권장한다. 웹 프레임웍을 사용하되, 이미 표준에 가까워진 Struts 프레임웍과 같은 대중적인 프레임웍을 사용하고 화면은 JSF 와 AJAX 로 구현하도록 한다.

웹 어플리케이션 개발은 새로운 전환점을 맞이 하고 있다.

이제 단순히 웹 브라우저 화면을 통해 정보를 주고 받거나, 하이퍼 링크를 통해서 화면의 이동을 요청하던 형태에서 벗어나 다양한 디바이스의 사용자는 보다 더 인터랙티브하고 동적인, 개발자 측면에서는 보다 쉽고 빠르게 구현하기를 바라고 있다.

이러한 RIA 를 구현하기 위해 솔루션을 도입한다든지, 새롭게 대두되는 기술을 이용하는 등 다양한 방법으로 접근이 가능하다. 어떤 기술을 적용할지는 구축할 대상과 요구사항에 따라 선택하여야 할 것이다.

참고문헌

[1] Forrester Research, Inc. " The X Internet" 2001
[2] Kevin Mullet " The Essence of Effective RIA" 2003
[3] IDC. Joshua Duhl " Rich Internet Applications" 2003
[4] Sun JavaOne " Rich Web Applications With Java™ 2 Platform, Enterprise Edition, J2EE™Platform and AJAX" 2005
[5] Gartner "Rich Internet Applications Are the Next Evolution of the Web" 2005
[6] Kito D. Mann " JavaServer Faces in Action" 2004
[7] Cay Horstmann " Core JavaServer Faces" 2004
[8] Sun JavaOne " Web FrameworkSmackdown" 2005