

포틀릿 기반의 인터랙티브 웹 서비스 구현

박재훈^o, 류영현, 시대근, 정영식, 정석태, 한성국
원광대학교 컴퓨터공학과

{pjh98^o, sdk124, toughryu, ysjeong, stjoung, skhan}@wonkwang.ac.kr

An Implementation of Interactive Web Services based on Portlets

Jaehun Park^o, DaeKeun Si, YeongHyeon Ryu, YoungSik Jeong, SuckTae Joung, SungKook Han
Dept of Computer Engineering, Wonkwang University

요 약

웹 서비스는 사용자의 요구사항과 기술의 발전에 따라 다양한 형태로 발전이 필요하다. 이러한 흐름에 따라 사용자 인터페이스의 개선을 목적으로 하는 인터랙티브 웹 서비스가 요구되고 있다. 본 논문에서는 사용자 인터페이스를 제공하는 포틀릿을 활용한 인터랙티브 웹 서비스 구현 방법을 제시한다. 포틀릿을 이용해서 표현과 처리를 분리할 수 있고, 포틀릿의 간단한 변경으로 다양한 사용자 인터페이스를 제공할 수 있다. 이로써 사용자는 별도의 프로그램 제작 없이 웹 서비스와 함께 사용자 인터페이스를 선정하여 원하는 형태의 결과를 볼 수 있다. 웹 서비스 응용시스템 구축시, 표현 계층의 설계 및 구현을 효과적으로 수행할 수 있다.

1. 서론

인터넷이라는 새로운 패러다임이 등장한 이 후 웹은 급속도로 발전하기 시작했다. 초기의 인터넷은 웹을 통해 정보를 수집하는 단계였다면 현재는 웹과 비즈니스를 통합하는 단계까지 이르고 있다. 더 나아가 비즈니스의 수행 방법을 변화시키며 새로운 비즈니스의 생성을 가능하게 하는 단계로 나아갈 것이다. 웹을 통해 할 수 있는 일들의 변화는 모든 기업의 비즈니스에도 변화를 가져와 기존의 비즈니스의 개념들의 많은 변화를 가져오고 있다.

초기의 웹 서비스는 비즈니스 로직을 처리한 후 서비스의 결과만 제공하는 형태이다. 하지만 웹 서비스는 사용자의 요구사항과 기술의 발전에 따라 다양한 형태로 발전이 필요하게 된다. 이러한 흐름에 따라 사용자 인터페이스의 개선을 목적으로 하는 인터랙티브 웹 서비스(interactive Web Services)가 필요하게 되었다. 본 논문에서는 사용자 인터페이스 서비스를 제공하는 포틀릿을 활용하여 인터랙티브 웹 서비스를 구현하는 방법을 제시한다. 사용자는 웹 서비스 선정과 함께 사용자 인터페이스 서비스를 제공하는 포틀릿을 활용함으로써, 웹 서비스 기반 응용 시스템을 단일 관점에서 효과적으로 구현할 수 있다.

2. Web Service 유형

웹 서비스는 비즈니스 로직의 처리결과 형태에 따라 프로그래마틱 웹 서비스(programmatic Web Services)와 인터랙티브 웹 서비스로 나뉜다. 프로그래마틱 웹 서비스는 단순한 처리결과와 메시지를 받게 되어, 사용자가 결과를 확인하려면 별도의 프로그래밍 작업이 필요하다. 반면에, 인터랙티브 웹 서비스에서는 웹 서비스 요청에 대한 결과뿐만 아니라 사용자 인터페이스를 포함해 사용자에게 보내므로 별도의 프로그래밍 없이 처리결과와 확인이 가능하다[1].

2.1 프로그래마틱 웹 서비스

서비스 요청자가 서비스 제공자에게 프로그래마틱 형태의 웹 서비스를 통해 서비스를 요청할 경우 사용자 인터페이스는 포함되지 않은 처리결과를 제공한다.

웹 서비스는 크게 사용자 인터페이스, 비즈니스 로직, 데이터로 나뉜다. 그림 1에서 서비스 요청자가 요청한 서비스에 대해 서비스 제공자가 사용자 인터페이스를 포함하지 않고 데이터와 비즈니스 로직만 처리결과로 넘겨준다. 이 결과를 이용해 서비스 요청자는 별도의 사용자 인터페이스를 작성해야 처리결과와 확인이 가능하다.

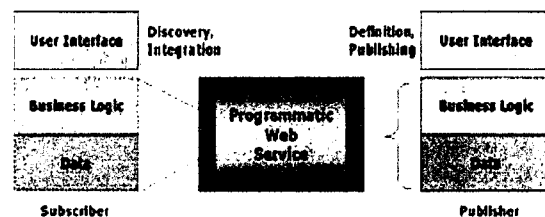


그림 1. 프로그래마틱 웹 서비스

사용자 인터페이스가 중요한 이유는 서비스 제공자가 넘겨준 결과 데이터는 레코드셋과 같은 물리적 저장소에 저장되어 있으므로 처리결과를 확인하려면 인터페이스가 필요하다. 그러므로 사용자 인터페이스가 없으면 서비스 요청에 대한 처리결과가 제대로 넘어왔는지 확인하기가 번거롭고 어렵다. 이런 단점을 보완해 사용자 인터페이스를 포함해 처리하는 인터랙티브 웹 서비스 개념이 등장한다[3].

2.2 인터랙티브 웹 서비스

서비스 요청자가 서비스 제공자에게 인터랙티브 형태의 웹 서비스를 통해 서비스를 요청할 경우 사용자 인터페이스가 포함된 처리결과를 제공한다.



그림 2. 인터랙티브 웹 서비스

그림 2에서 서비스 요청자가 요청한 서비스에 대해 서비스 제공자가 사용자 인터페이스가 포함된 결과를 넘겨준다. 서비스 요청자는 이 결과에 별도의 프로그램 제작 없이 바로 처리 결과를 확인할 수 있다. 서비스 제공자 측에서 사용자 인터페이스를 포함해서 제공하므로 서비스 요청자는 프로그램 제작에 드는 비용이나 시간을 절약할 수 있고, 결과 데이터의 유형에 따른 사용자 인터페이스의 변경으로 인한 유지보수 문제도 해결된다[3].

3. 인터랙티브 웹 서비스 구현 방법

2001년 중반 WSIA와 WSRP 등장한 후 WSIA는 포탈 개발업체인 Epicentric에 의해 WSUI로 IBM에 의해 WSXL로 나누어져 퍼지게 된다. 많은 포탈 개발업체들은 포탈 툴킷에 대한 동일한 문제에 봉착하게 된다. 이 문제점은 리모트 서비스에 대해 각각의 프로그램을 제작해야 한다는 점이다. 이를 해결하기 위해 다양한 방법이 제시된다[4].

3.1 인터랙티브 웹 서비스 표준안

■ Web Service User Interface

WSUI는 인터랙티브 웹 서비스를 위한 하나의 마크업 언어이다. WSUI는 표준기술 기반이므로 쉬운 구현이 가능하고 다양한 표현 언어를 지원한다. 완전한 서버 중심의 독립적인 프로그래밍 언어를 사용한다. WSUI 컴포넌트를 생성할 경우 별도의 개발 절차 없이 가능하다[5].

■ Web Services for Remote Portals

WSRP는 interactive, presentation 지향 웹 서비스를 표준 포탈에 간단하게 이식이 가능하다. 또한 포탈과 웹 서비스 내에서 개념이나 데이터가 다른 표준으로의 변환을 허용한다. WSRP는 WSDL을 정의하는 interface 역할을 하며 등록된 WSRP를 검색하고 출판하기 위해 자동으로 메타 데이터를 작성한다.

■ Web Service Experience Language

WSXL은 인터넷상에서 사용자에게 체험을 제공하는 어플리케이션 중심이며, 인터랙티브 웹 어플리케이션을 위한 웹 서비스 중심의 컴포넌트 모델이다.

WSXL은 다음의 두 가지 주요 목표를 달성하기 위해 설계되었다. 하나는 회사들이 여러 배포 채널을 통해 양방향 웹 어플리케이션을 전달할 수 있도록 하는 것이고, 다른 하나는 웹에서 다른 양방향 어플리케이션들을 응용하여 새로운 서비스에

어플리케이션이 만들어질 수 있도록 하는 것이다. 이 목표를 달성하기 위하여, 모든 WSXL 컴포넌트 서비스는 라이프 사이클 관리를 위하여 기초 운영의 설정을 이행하고 사용자가 입력하는 것을 받아들이며 프레젠테이션 마크업을 제작한다.

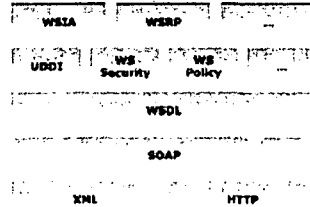


그림 3. WSIA와 WSRP를 이용한 Web Service 구조

■ WSUI, WSRP, WSXL의 관계

WSUI, WSRP, WSXL의 공통점은 모두 인터랙티브 웹 서비스를 지향한다는 점이다. 초기의 인터랙티브 웹 서비스는 WSIA와 WSRP를 이용했다. 기존의 웹 서비스 계층의 상단에 WSIA와 WSRP가 존재해 인터랙티브 웹 서비스의 구현이 가능했다. 그림 3은 기존의 웹 서비스 계층에 WSIA와 WSRP가 적용된 계층 구조이다.

WSUI는 웹 서비스와 HTML의 Table Tag, CSS 등의 인터랙티브 요소를 포함하고 있는 마크업 언어이다. 즉, 웹 서비스 계층이나 HTTP 등과 같은 계층을 포함하고 있는 언어이므로 기존의 웹 서비스 계층 구조를 대부분 포함하고 있는 것이다. 이와 비교되는 WSXL은 애플리케이션 중심의 웹 서비스이다. 즉, WSXL도 웹 서비스 계층을 포함하고 있는 형태이다. 개념적으로 이해했을 때 WSUI를 이용한 웹 서비스와 WSXL을 이용한 웹 서비스로 나누어 생각해 볼 수 있다[2].

3.2 포틀릿 기반 인터랙티브 웹 서비스

포틀릿과 웹서비스는 미리 정의된 UI가 각각 적용된다. UI는 CSS 파일에 클래스로 정의되어 있고 포틀릿과 웹서비스를 호출하는 페이지에서 UI 클래스를 지정한다.

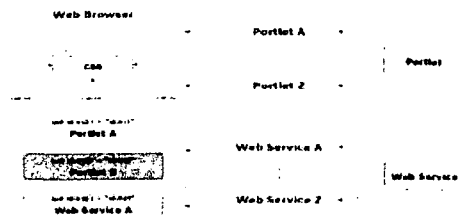


그림 4. 인터랙티브 웹 서비스 Page 구성

그림 4는 구현 웹 페이지의 구성이다. 페이지의 구성 순서는 다음과 같다.

1. CSS 클래스 파일의 포틀릿 Header UI 호출
2. 로컬 디스크의 포틀릿 호출
3. CSS 클래스 파일의 포틀릿 Bottom UI 호출

각각의 포틀릿은 UI로 씌워지는 형태로 UI 클래스 변수에 의

해 변경된다. 자세한 포틀릿 구성은 다음과 같다.

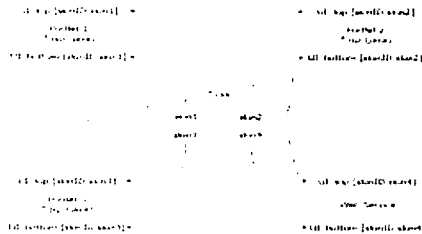


그림 5. 포틀릿의 UI 적용

위의 포틀릿은 컨테이너 개념의 웹 페이지를 구성하고 웹 서비스 또한 마찬가지다. 각각의 포틀릿과 웹 서비스는 skinID라는 값에 의해 UI가 결정되고 UI는 UI_top과 UI_bottom으로 나누어 포틀릿 또는 웹 서비스를 감싼다. 이렇게 UI가 적용된 포틀릿은 미리 정해놓은 디자인 템플릿에 맞게 웹 페이지를 구성한다.

4. 포틀릿 기반 인터랙티브 웹 서비스 구현

인터랙티브 웹 서비스의 구현은 3개의 포틀릿과 1개의 웹 서비스를 기반으로 웹 페이지에 UI를 적용시키는 간단한 프로그램을 JSP로 구현했다. 포틀릿은 아날로그 시계, 계산기, 지뢰 찾기 게임으로 기존에 자바 또는 자바 스크립트 형태로 구현된 소스이고 웹 서비스는 실시간으로 제공되는 뉴스 서비스이다. 이들 포틀릿, 웹 서비스의 UI는 CSS에 선언한 클래스를 이용해 적용한다. UI는 클래스 변수의 수정을 통해 미리 CSS에 작성해 놓은 디자인으로 변경되어 적용된다.

```
<%
SkinID = "Skin3"; // UI 설정값
%>
<%@ include file="/Common/box_top.jsp" %> // Top UI
<table border="0" width="100%">
<tr>
<td align="center">
< %>< include file="framesweeper.jsp" %> // 지뢰찾기 포틀릿
</td>
</tr>
</table>
<%@ include file="/Common/box_bottom.jsp" %> // Bottom UI
```

그림 6. 포틀릿 호출과 UI 적용 소스의 예

위의 그림은 지뢰찾기 포틀릿을 호출하고 UI가 적용되는 소스이다. CSS 파일에 클래스로 정의된 UI 클래스 값을 지정하면 Top UI와 Bottom UI가 UI 클래스에 지정된 디자인으로 적용되어 사용자의 웹 브라우저에 적용된다. 다른 포틀릿이나 웹 서비스도 위와 동일한 루틴으로 웹 페이지를 구현하면 그림 7과 같은 UI가 적용된 웹 페이지를 확인할 수 있다.

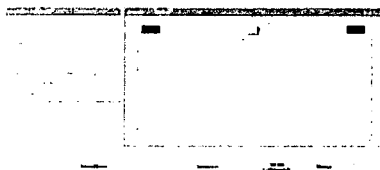


그림 7. 프로그램 실행 화면

5. 결론

웹 서비스는 일반적으로 요청 서비스의 비즈니스 로직을 처리해 결과만을 제공해주는 것이다. 문제는 서비스 제공자가 처리한 결과에 대한 형식이다. 처리결과만 제공하므로 결과를 받은 서비스 요청자는 결과를 확인하려면 별도의 프로그램을 작성해 결과 데이터를 표현해야 한다. 하지만 서버가 결과를 확인할 수 있는 형태로 서비스를 제공한다면 서비스 요청자는 아무런 조작 없이 결과를 확인할 수 있다. 이를 가능하게 하려는 노력은 WSUI, WSXL과 WSRP를 통해 알 수 있다.

WSUI는 인터랙티브 웹 서비스를 위한 하나의 마크업 언어이다. XML 기반으로 UI를 처리할 수 있는 Element 가 포함되어 있다. WSXL은 특수한 웹 서비스로 웹 서비스를 수행할 수 있는 별도의 처리 로직이 포함되어 있다. 기존의 비즈니스 로직만 처리하던 웹 서비스가 UI를 생성하고 제공할 수 있는 로직이 포함된 것이다. 본 논문에서는 이러한 개념을 간단한 구현을 통해 알아왔다. 포틀릿을 이용해서 표현과 처리를 분리할 수 있고, 포틀릿의 간단한 변경으로 다양한 사용자 인터페이스를 제공할 수 있다. 이로써 사용자는 별도의 프로그램 제작 없이 웹 서비스와 함께 사용자 인터페이스 서비스를 선정하여 원하는 형태의 결과를 볼 수 있다. 그러나 인터랙티브 웹 서비스의 문제는 UI가 제한적이고 고정적인 형태이므로 사용자의 선택은 국한될 수밖에 없다. 이 문제의 해결책은 다양한 사용자 인터페이스를 서비스 제공자가 제작하고 서로 공유해 서비스 요청자에게 제공한다면 가능하다. 인터랙티브 웹 서비스는 서비스 요청자의 편의와 효율성을 만족시켜주는 충분한 조건을 갖춘 웹 서비스가 될 것이다.

감사의 글

본 연구는 학술진흥재단 기초과학연구사업 중 지방 연구중심대학 육성사업인 헬스케어 기술개발사업단의 지원에 의해 수행되었으며 이에 감사드립니다.

6. 참고문헌

- [1]IBM Corp, Microsoft Corp, "Web Services Framework", W3C Workshop on Web Services, San Jose, USA, 2001.
- [2]Claus Brabrand, Anders Møller, Anders B. Sandholm, Michael I. Schwartzbach "A Runtime System for Interactive Web Services", Amsterdam, Netherlands: 1999
- [3]Web Collage "Interactive Web Services ", Architecture White Paper, 2003
- [4]Elion Reshef "Building Interactive Web Services with WSIA & WSRP", Web Service Journal, December 2002
- [5]WSUI org "WSUI Specification", June 2001
- [6]박범대, "SW개발 패러다임의 변화 '웹서비스' - 웹서비스 발전방향과 전망", 경영과 컴퓨터, 2003년 9월