

웹 프로젝트에서 디자인과 프로그램의 분리에 관한 연구

정강용*, 김원중**, 허영남**

*순천제일대학 컴퓨터과학부

**순천대학교 컴퓨터과학과

jung740@hanmail.net, kwj@sunchon.ac.kr, hyn@sunchon.ac.kr

A Study on Separating Design from Program on Web Project

Gang-Yong Jung*, Won-Jung Kim**, Yeong-Nam Heo**

*Dept of Computer Science, Suncheon First College

**Dept of Computer Science, Suncheon National University

요 약

웹 프로젝트는 다양한 언어와 기술들이 결합되어 매우 복잡하고, 개발 및 유지보수의 생산성이 낮은 분야이다. 특히, 화면의 디자인과 프로그램 코드가 혼재되어 있다는 것이 공동개발과 이해를 가장 어렵게 한다. 웹 기반 시스템을 개발하기 위해서는 사용자 인터페이스 디자이너가 먼저 화면 레이아웃을 설계한 후 그 화면의 레이아웃 소스를 가지고 다시 프로그램 코드 개발자가 코딩을 하여 프로젝트를 완성한다. 유지 보수를 하는 경우에도 이와 비슷한 형태로 작업이 진행된다. 이러한 웹 기반의 소프트웨어 개발 방법은 JSP의 태그 라이브러리와 같은 다양한 형태로 제안되었으나 실제로 적용하는데 여러 가지 문제점이 있으며, 기존의 C/S 기반의 소프트웨어 개발에 비해 생산성 측면에서도 매우 좋지 않다. 본 논문에서는 웹 환경에서 운영되는 정보 시스템의 개발 과정에서 발생하는 디자인(사용자 인터페이스)과 프로그램을 분리하기 위한 방법론들의 문제점들을 제시하고, 효율적인 분리 방법론이 갖추어야 할 조건들을 제시하였다.

1. 서론

HTML을 기반으로 하는 웹 개발 환경에서는 4세대 언어에서 이미 해결한 디자인(사용자 인터페이스)과 프로그램 코드를 구분하는 효과적인 방법을 찾지 못하여 웹 기반의 소프트웨어 개발에 장애가 되고 있다. 따라서 웹 기반의 소프트웨어를 개발하는 경우 사용자 인터페이스 디자이너와 프로그램 코드 개발자간의 효율적인 의사소통과 풍부한 협업 개발 능력이 필요하다. 그러나 현재 웹 기반의 소프트웨어가 체계적이고 효율적인 개발 방법을 이용해 진행되기보다는 경험에 의한 개발 방법에 의존하고 있다. 이러한 문제점은 웹 기반의 소프트웨어를 개발하는데 있어서 비효율성이 증가하게 되는 요인이 되고 있다[5].

웹 기반 프로젝트와 4GL을 이용한 프로젝트를 소프트웨어 공학적으로 분석하면 웹 기반 프로젝트의 개발 생산성이 훨씬 낮다. 웹 기반 프로젝트의 개발

생산성이 저조한 이유는 개발 방법 자체가 복잡하며, 사용자 인터페이스와 프로그램 코드를 분리할 수 있는 표준화된 방법이 존재하지 않기 때문이다.

본 논문에서는 웹 환경에서 운영되는 정보 시스템의 개발 과정에서 발생하는 디자인과 프로그램 코드를 분리하기 위한 방법론들의 문제점들을 제시하고 효율적인 분리 방법론이 갖추어야 할 조건들을 제시하였다.

2. 관련연구

2.1 레거시 시스템에서의 분리 방법

IBM 3270과 같은 메인 프레임 기반의 텍스트 환경에서도 사용자 인터페이스와 프로그램 코드를 분리하기 위한 노력은 있었다. 유능한 프로그래머는 모듈화나 간단한 사용자 라이브러리를 만들어 사용하였으며 점점 상용화된 개발 도구들에 의해서 보완되었다. IBM 메인프레임의 CUI기반의 사용자 인터페이스 환경에서 Message Format Services라는 시

시스템을 이용하여 화면의 입출력 부분을 별도의 파일로 작성하여 컴파일하고 코블이나 PL/I으로 제작된 프로그램은 API를 이용하여 화면 입출력을 용이하게 설계, 개발할 수 있도록 하였다[6,8].

2.2 웹 환경에서 분리 방법

HTML이 사용자 인터페이스를 위한 화면 레이아웃을 설계하는데 사용됨으로서 웹 개발 초기엔 프로그래머가 디자인에 관련된 마크업 언어를 이해하고 코딩할 수 있어야 했으며, 디자이너도 프로그래머가 사용하는 프로그래밍 언어를 알아야 했다. 또한, 4GL 언어와는 다르게 사용자 인터페이스 관련 디자인이 상당히 중요한 부분을 차지하게 되어 기존의 RAD(Rapid Application Development) 틀에서 제공되던 사용자 인터페이스 관련 레이아웃을 사용할 수가 없게 되었다.

이런 문제를 해결하기 위하여 다양한 방법들이 제안되었는데 서버 사이드 스크립트 기반의 템플릿 방식, 4GL 방식, CMS 방식, CBD 기반의 프레임워크 4가지로 분류할 수 있으며 각각의 특징과 문제점은 다음과 같다.

2.2.1 서버 사이트 스크립트 기반의 템플릿 방식

서버 사이트 스크립트 기반의 템플릿 방식은 웹 상에서 응용 프로그램을 개발하는데 필요한 스크립트 언어의 특징을 활용하는 방식이다.

WWW(World Wide Web)에서 애플리케이션을 개발하는 기본 기술인 CGI 방식으로 웹 애플리케이션을 개발하는 경우 프로그램 코드와 사용자 인터페이스 부분이 전혀 분리되지 않고 하나의 비즈니스 로직 개발 코드 안에 포함된다. 이런 문제를 해결하기 위하여 제안된 것이 서버 사이트 스크립트로서 대표적인 서버 사이트 스크립트는 ASP(Active Server Page), JSP(Java Server Page), PHP(Personal Home Page Tools) 등이 있다. 서버 사이트 스크립트는 전체 코드 안에 프로그램 코드와 사용자 인터페이스 관련 코드 부분을 분리하여 표현할 수 있다. 그러나 별도의 코드로 완벽하게 분리되는 것이 아니고 동일한 코드 안에서 단순히 프로그램 코드와 사용자 인터페이스 관련 부분의 영역만을 구분한다.

이 방식은 프로그램 코드 개발자의 부담을 줄여 주었으나 사용자 인터페이스와 프로그램 코드를 완전히 분리하지는 못한다. 이런 문제를 해결하기 위해 ASP에서는 DLL(Dynamic Link Library) 파일, JSP에서는 Java Beans, PHP에서는 Smarty(the compiling template engine for PHP)와 같은 템플릿 기법을 도입하여 프로그램 코드와 사용자 인터페

이스 부분을 분리할 수 있다. 그러나 특정 스크립트 언어에 종속되며, 개발자와 디자이너가 템플릿 기법에 대한 사전 지식이 있어야 하며, 개발자나 디자이너 둘 중에 한 파트의 작업자가 다른 파트의 작업자에 비해 더 많은 역할을 수행하여야 한다[1,7].

2.2.2 4GL 방식

4GL 언어가 애플리케이션 분야에 폭 넓게 사용된 것은 화면 디자인을 하는데 필요한 여러 가지 컴포넌트들을 정형화시킨 후, 그 컴포넌트에 속성을 부여하여 개발자별로 손쉽게 GUI 기반의 응용 프로그램을 개발할 수 있기 때문이다. 이 방식은 Microsoft에서 제안한 웹 폼, SUN에서 제안한 JSF(Java Server Faces) 방식이 있다. 웹 폼이나 JSF는 Visual Basic이나 Delphi 등과 같은 4GL 개발 틀에서 사용하던 방식과 비슷하게 정형화된 사용자 인터페이스 컴포넌트를 활용하여 사용자 인터페이스 디자이너의 역할을 최소화하였다. 4GL 방식의 기법을 활용한 방식은 아직은 개발이 진행 중이기 때문에 폭넓게 사용되고 있지 않다[9].

2.2.3 CMS(Contents Management System) 방식

웹 상에서 응용 프로그램을 개발하여 사용하는 분야는 다양하며 쇼핑몰의 경우 상품을 전시하고 판매하는 일반적인 기능 특성을 가지며, 커뮤니티 분야는 회원들 간에 정보를 교환할 수 있도록 하는 기능, 교육 분야는 강의실 관리나 학생관리 같은 공통적인 특성이 존재한다. 이런 각각의 분야별로 필요한 응용 프로그램들을 미리 개발하여 템플릿 형태로 저장해 놓은 후 사용자 요구에 따라 사용자 인터페이스 디자인만을 변경하는 것이 CMS 방식이다. 이 방식의 경우 개발자의 역할이 많이 축소되고 사용자 인터페이스 디자이너가 주축이 될 수 있다는 장점이 있다. 단점으로는 특정 분야에 종속적으로 개발되었기 때문에 확장성이 미흡하며, 사용자의 다양한 요구 사항을 만족시키지 못한다는 점이다[3,4].

2.2.4 CBD 기반의 프레임워크

웹 상에서 개발되는 수많은 응용 프로그램들의 가장 큰 문제점은 재사용성이 부족하다는 점이다. 즉 웹 응용 프로그램들은 개발되고 난 후에 다른 시스템에서 재사용하기가 어렵다. 이런 문제점을 해결하기 위하여 웹에서 응용 프로그램을 개발하는데 필요한 각각의 모듈들을 미리 개발해 놓고 필요하게 되면 약간의 수정을 가하거나 그대로 이용하는 CBD 기반의 개발 방식을 사용하는 것이다. 이 방식의 예로는 Microsoft사의 .NET이나 SUN의 J2EE가 있다.

3. 디자인과 프로그램의 분리

3.1 웹 개발 프로젝트의 특성

기존에 사용되어온 메인프레임 환경이나 C/S기반의 개발 환경에서의 정보 시스템 프로젝트가 웹 개발 프로젝트에 적용할 수 없거나, 개발 생산성이 저조한 이유는 웹 프로젝트 자체가 여러 기술들이 복합되어 매우 복잡하며, 사용자 인터페이스와 프로그램 코드를 분리할 수 있는 표준화된 방법이 존재하지 않고, 다음과 같은 웹 개발 프로젝트의 특수성 때문이다.

첫째, 웹 개발 프로젝트의 기반이 되는 HTTP 프로토콜의 가장 큰 특징은 비연결성이다. 즉, 웹 브라우저가 서버에 연결하고 정보를 요청하면, 서버는 연결된 브라우저에 처리된 정보를 제공하고 연결이 끊어진다는 것이다. 이런 구조에서는 지속적이고 안정적인 상태 지속을 할 수 없기 때문에 쿠키나 세션과 같은 별도의 방법을 이용하여 상태를 지속할 수 있으나 이 방법은 보안 측면이나 안정성 측면에서 보완해야 할 점이 많다[6].

둘째, 시스템 구성의 복잡성에 있다. C/S 기반의 4GL 툴로 개발된 소프트웨어의 경우 사용자 컴퓨터의 클라이언트 프로그램과 서버의 데이터베이스만이 작업을 수행하지만 웹 기반의 프로젝트의 경우 클라이언트 프로그램인 웹 브라우저와 서버의 데이터베이스, 그리고 서버의 데이터베이스를 가공하여 사용자의 클라이언트 프로그램에 데이터를 전송하는 웹 애플리케이션이 필요하다. 이런 복잡한 구조의 시스템의 효율적인 구성을 위해서는 기존의 개발 방법보다 훨씬 더 많은 노력을 요구하게 된다[5].

셋째, 기존의 4GL 툴은 표준화된 사용자 인터페이스 컴포넌트를 구현하여 별도의 사용자 인터페이스 디자이너가 없어도 애플리케이션을 구현할 수 있었으나 웹 기반 애플리케이션은 표준화된 사용자 인터페이스 컴포넌트가 없기 때문에 개발 과정에서 사용자 인터페이스 디자이너의 지원이 필수적이다. 사용자 인터페이스 디자인과 프로그램 코드의 동시 개발에 대한 직관적인 개발 방법이 존재하지 않기 때문에 사용자 인터페이스 디자이너의 작업 후 프로그램 개발자의 작업이 진행되며, [그림1]과 같이 디자이너와 프로그래머가 하나의 소스를 가지고 작업을 하고, 소스에 사용자 인터페이스 디자인과 프로그램 코드가 혼재되어 있기 때문에 가독성이 저하되고, 수정 과정에서 난이도가 높아지며 소스의 안정성 유지에도 문제가 있다.

웹 기반 애플리케이션의 비연결성에 대한 제약점은 플러그인이라는 기술을 활용하여 극복되었으며 웹 기반 시스템이 가지는 복잡성은 WAS(Web

Application Server) 등의 표준화된 개발 방법을 사용하면서 극복할 수 있었다. 그러나 사용자 인터페이스 디자인과 프로그램 코드의 분리에 대한 연구는 지금도 진행 중이며 아직 완벽한 해결 방법이 없다 [5,8].

<pre><td width=30><table width=30 cellspacing=0 cellpadding=0 border=0></td> </tr></pre>
<pre><%-- Board의 articles에 저장된 페이지 리스트를 차례로 출력한다 --%> <% if (board.getNumberOfCurrentArticles() == 0) { %></pre>
<pre><tr bgcolor="#FFFFFF"> <td colspan="<%= numberOfFields %>" align=center> 등록된 글이 없습니다. </td> </tr></pre>
<pre><% } else { for (int i = 0; i < board.getNumberOfCurrentArticles(); i++) { %></pre>
<pre><tr bgcolor="#FFFFFF"> <% board.Article tuple = board.getArticle(i); int index1 = tuple.getIndex1();</pre>

[그림 1] 디자인과 프로그램 코드가 혼재된 JSP 소스

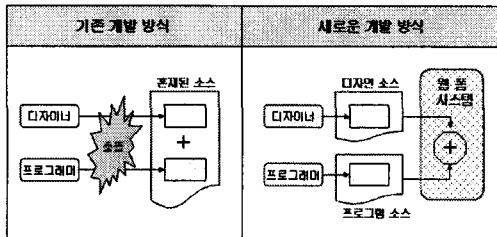
3.2 기존 방법들의 문제점

앞장에서 언급한 기존의 방법들에는 공통적인 단점들이 있으며, 그 내용은 다음과 같다. 첫째, 각 벤더들은 각자 기존의 웹 운영과 개발 도구에 종속된 방법들을 제공하기 때문에 다른 웹 프로그래밍 언어나 웹 운영 환경과 호환이 되지 않는다. 둘째, 각 환경마다 기술의 습득이 어려우며 다른 개발 환경 및 운영 환경에 적용하기가 어렵다. 셋째, 기존의 디자인 전문 툴을 모두 활용하기가 어렵다. 넷째, 디자인과 프로그램 코드의 분리를 위해 Heavy Platform을 적용하여야 하므로 도입과 유지보수 비용이 많이 든다. [표1]은 기존 방법의 문제점을 항목별로 분류했다.

[표 1] 기존 방법의 문제점

단 점	내 용
시스템 환경 종속	.Net 플랫폼과 J2EE 등 각기 다른 환경 사용
웹 개발 언어 종속	JSP, ASP, PHP 등 언어에 따라 사용 툴이 상이함
상이한 기술 습득	Beans, Web Form, Smarty 등 상이한 전문 기술 필요
기타 문제점	디자인과 로직 분리만을 위해 Heavy Platform 사용

따라서 이러한 문제점들을 해결하기 위해서는 [그림2]와 같이 사용자 인터페이스 디자이너와 프로그램 코드 개발자를 분리시켜 사용자 인터페이스 디자이너와 프로그램 코드 개발자와의 의견 충돌이나 소스 코드 버전 관리를 수월하게 할 수 있는 방안이 절실히 요청된다. 즉, 사용자 인터페이스 디자이너가 사용자 인터페이스 관련 디자인이 끝난 후 프로그램 코드의 개발이 진행되는 문제를 해결하여, 동시에 개발을 진행할 수 있도록 함으로써 소프트웨어 생산성을 향상시킬 수 있는 방법론이 필요하다.



[그림2] 개발 방식 비교

3.3 디자인과 프로그램 분리를 위한 기본 조건

사용자 인터페이스와 프로그램 코드의 분리를 위한 개발 방법론이 갖추어야 할 기본적인 기능들은 다음과 같다.

첫째, 플랫폼에 독립적이어야 한다. 마이크로소프트사의 .Net이나 Sun사의 J2EE와 같은 플랫폼에 종속되지 않고 어느 플랫폼에나 쉽게 적용될 수 있어야 한다. 웹 애플리케이션을 개발하는데 있어서 특정 플랫폼에 종속된다는 것은 웹이 추구하는 기본 방향과 일치하지 않으며, 특정 기술에 종속되는 경우 개발 시스템 자체에 제약이 된다.

둘째, 기존의 디자이너와 프로그래머가 쉽게 사용할 수 있도록 하여야 한다. 즉, 완전히 새로운 디자인 도구를 사용하거나 새로운 언어를 습득하지 않고, 약간의 API와 디자인 규칙만으로 사용할 수 있어야 한다.

셋째, 기존의 웹 개발 프로그래밍 언어에 종속적이지 않고 모든 언어에서 동일한 방법으로 사용할 수 있어야 한다. JAVA 환경에서 개발하던 프로그래머가 PHP 환경에서 새로운 시스템을 개발하려 할 때 JAVA에서 사용하던 방법을 그대로 사용할 수 있어야 한다.

넷째, 현재 널리 사용되고 있는 보편적인 디자인과 프로그램 도구를 계속 사용할 수 있어야 한다. 다시 말해서 제안된 방법이 HTML 태그 중에 "<TABLE>"을 사용할 수 없다는 식의 제약 조건을

가져서는 안 된다.

여섯째, 사용자 인터페이스 디자이너와 프로그램 코드 개발자가 동시에 작업 가능하도록 하여야 한다. Macromedia사의 드림위버는 사용자 인터페이스 디자이너와 개발자가 같은 화면을 통해 개발 작업이 가능하지만 동시에 개발을 진행할 수 없다는 단점은 해결하지 못하였다.

4. 결론 및 향후 연구

웹을 기반으로 하는 소프트웨어는 사용자 인터페이스 디자이너의 비중이 증가되면서 프로그램 코드 개발자와 사용자 인터페이스 디자이너의 개발 경험과 원활한 의사소통이 필요조건이 되었다. 이런 문제점을 해결하기 위해서는 웹 기반의 소프트웨어에서 사용자 인터페이스 부분과 프로그램 코드 부분을 분리하여 서로의 간섭을 최소화시키는 것이 관건이지만 현재까지 제안된 방법들은 여러 가지 문제점들을 가지고 있다.

본 논문에서는 웹 프로젝트 개발에서 사용자 인터페이스 디자이너와 프로그램 코드 개발자를 분리하기 위한 기존의 방법들을 소개하고, 각 방법들이 가지는 문제점들에 대해 설명하였다. 그리고 이들 문제점들을 해결하기 위한 새로운 방법론이 갖추어야 할 기본적인 조건과 기능을 제시하였다. 현재 이를 기반으로 디자인과 프로그램 코드를 효율적으로 분리할 수 있는 웹 폼 시스템에 대한 연구를 계속 진행하고 있다.

참고문헌

- [1]ASP, <http://www.microsoft.com>
- [2]Anders Kristensen, "Template Resolution in XML/HTML," HP Labs Technical paper, 1999
- [3]Engin K. and Clemens K., "MyXML: An XML based template engine for the generation of flexible web content," Technical Univ. of Vienna, 2001
- [4]ITdata, <http://www.itdata.co.kr/column/200212/tech/tr08.asp>
- [5]Jessica Burdman, "Collaborative Web Development : Strategies and Best Practices for Web Teams," Addison Wesley Longman, 2000
- [6]M. Schranz. "Management process of WWW services: an experience report." SEKE '97, 1997
- [7]PHP, <http://smarthy.php.net/>
- [8]The history of Computing, <http://ei.cs.vt.edu/~history>
- [9]Web forms, <http://msdn.microsoft.com/library/default.asp>