

# MVC 기반의 웹 컴포넌트 컨테이너 시스템에 관한 연구

유주현<sup>0</sup> 황인문 김원중  
순천대학교 컴퓨터학과  
kwj@sunchon.ac.kr

## A Study on Web Component Container System Using MVC

Juhyun Yoo<sup>0</sup> Inmoon Hwang Wonjung Kim  
Dept. of Computer Science, Suncheon National University

### 요 약

대부분의 웹 기반의 개발 환경에서는 웹 디자인이 변경되는 경우 그에 해당하는 개발 코드를 수정하는 방법을 이용하여 개발한다. 이와 같은 개발 방식은 웹 프로그램의 개발 생산성 및 개발된 코드의 재사용성을 저하시키는 문제를 유발하였다. 그래서 웹 디자인과 개발 코드를 분리하기 위한 다양한 방법들이 제안되었으며 활용되어 왔으나 미흡했다. 본 논문에서 설계, 구현한 MVC기반의 웹 컴포넌트 시스템을 적용하는 경우 설계 단계에서부터 웹 디자인과 개발 코드를 쉽게 분리할 수 있도록 하여 웹 프로그램의 개발 생산성 및 개발 코드의 재사용성을 향상시켰다.

### 1. 서 론

현재 웹 기반의 개발 환경에서의 전통적인 개발 방법은 HTML과 JSP, PHP, ASP와 같은 서버 사이드 스크립트 개발 언어, 그리고 웹 디자이너가 설계한 화면 템플릿을 조합하는 형태로 진행된다. 기존의 C/S(Client/Server) 개발 환경과의 큰 차이점은 C/S환경에서는 별로 중요하지 않았던 화면 디자인이 웹 환경에서는 매우 중요한 위치를 차지한다는 것이다. 이와 같은 특징은 화면 디자인이 변경될 때마다 해당 개발 코드가 변경되어야 한다. 그래서 웹 기반 응용 프로그램의 개발 생산성 및 개발된 코드의 재사용성을 저하시켰다[1,2]. 이 문제를 해결하기 위하여 JAFAR, 4GL 방식등의 다양한 방법들이 제안되었으나 단순히 사용자 인터페이스와 개발 코드를 분리시키는데 중점을 두는 방식이 대부분이었다[2]. 그러나, 본 논문에서 제안한 MVC기반 웹 컴포넌트 컨테이너 시스템은 사용자 인터페이스와 개발 코드의 형식적인 분리보다는 개발 단계부터 각각의 개발 모듈들을 패턴으로 지정하고 지정된 각각의 패턴을 컴포넌트화 시켜 디자인 부분과 개발코드를 매치시키는 작업을 수행하여 디자인 부분과 개발 코드 분리뿐 아니라 개발 코드의 재사용성 및 개발 생산성을 향상시킬 수 있는 방안을 제시하였다.

### 2. 관련연구

#### 2.1 MVC

MVC는 소프트웨어 설계에서 사용되는 개발 방법론으로서

Model/View/Controller의 줄임말이다. MVC는 사용자와 상호 작용하는 소프트웨어를 설계하는 과정에서 다음과 같이 세 가지 요소 형태로 분리 할 수 있다[3].

Model은 응용 프로그램의 내부 상태, 즉 논리(Algorithms)나 자료(Data)와 같은 형태를 의미한다. View는 사용자에게 Model을 어떻게 보여주는 규정이며, 하는 일의 표현 방법이라 할 수 있다. Controller는 사용자의 입력(Input)과 응용 프로그램 간의 상호작용을 의미한다[3].

MVC를 개발환경에 적용하는 경우 각각의 소프트웨어를 컴포넌트 형태로 개발할 수 있다는 장점이 있다. SUN에서 제안한 Java Beans도 MVC 형태로 개발된 것이다 [3,4].

#### 2.2 웹 환경에서의 C/S 개발 방법론의 적용

C/S기반의 개발 환경이 웹 기반 소프트웨어보다 생산성과 재사용성이 뛰어난 이유는 View와 관련된 부분이 웹 개발 환경에 비해 쉽게 분리가 될 수 있다는 것이다. 다시 말해 View와 관련된 부분을 별도의 컴포넌트 형태로 저장한 후 개발자가 필요한 경우에 해당 컴포넌트를 호출하여 View부분을 구현할 수 있다는 것이다. 이런 C/S 환경의 장점을 수용하기 위하여 제안된 것이 자바 진영의 JSF(Java Server Faces)와 마이크로소프트사의 WebForm등이다.

#### (1) JSF

JSF는 웹 기반의 개발환경에서 가장 큰 문제점인 사용자

인터페이스 관련 부분을 처리하기 위하여 컴포넌트 형태로 작성한 후, 각각의 컴포넌트에 속성을 부여하는 방식으로 사용자 인터페이스 개발을 쉽게 할 수 있도록 제공하고 있다. 즉, 4GL(Fourth-Generation Language) 개발툴에서 사용하던 방식과 유사한 정형화된 사용자 인터페이스 컴포넌트라고 할 수 있다. 장점으로는 개발자가 웹 환경에서 사용자 인터페이스와 관련되어 다양한 기능과 표현이 가능 하지만 단점으로는 웹 디자이너가 웹 응용 프로그램 개발 능력이 있어야 하는 문제가 있다[2,5].

(2) WebForm

마이크로소프트사에서 개발한 사용자 인터페이스 컴포넌트로서 JSF와 비슷하다. WebForm의 가장 큰 장점은 웹 클라이언트 부분에 해석기가 불필요하다는 점이다. 서버에서 해석 처리되기 때문에 기존의 Win32를 이용한 어플리케이션 방식의 Window Form과 큰 차이가 있다. WebForm을 활용하는 경우 사용자의 다양한 요구는 충족할 수 있다는 장점이 있다[2,6].

3. MVC기반의 웹 컴포넌트 컨테이너의 설계 및 구현

3.1 기존 웹 개발 방법 및 문제점

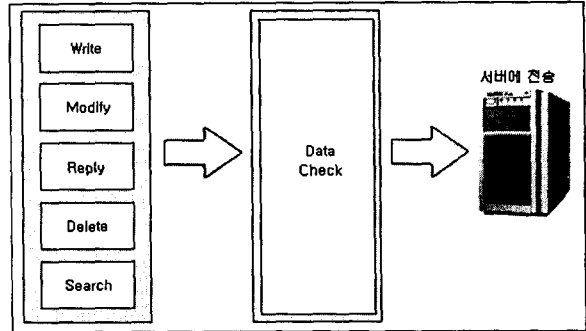
웹 기반의 전형적인 개발 방법은 HTML과 JSP, ASP, PHP 등과 같은 서버 스크립트 언어 및 웹 디자이너가 설계한 사용자 인터페이스 템플릿으로 구성된다. HTML코드와 서버 사이드 스크립트 언어가 공존하는 개발 형태는 HTML이 가지는 한계로 인하여 디자인이 결정되는 형태에 따라 코드 개발이 이루어 졌다. 이런 개발 방법은 웹 디자이너와 웹 프로그래머간의 충돌이 빈번하게 발생되었으며, 소프트웨어의 생산성 및 개발 코드의 재사용성에서도 문제가 많았다.

이러한 문제점을 극복하기 위하여 레거시 시스템에서의 MFS(Message Format Service) 시스템, 서버 사이드 스크립트언어의 템플릿 방식, WebForm과 JSF와 같은 4GL 방식, JAFAR(Javascript Array For All Relation) 등의 다양한 설계 방법이 연구되으나, 사용자 인터페이스와 개발 코드의 분리에 중점을 두는 방식이 많았다. 본 논문에서는 MVC 기반의 웹 컴포넌트들을 설계, 구현하여 웹 기반 소프트웨어를 설계 단계에서부터 사용자 인터페이스와 개발 코드를 분리시켜 소프트웨어 개발의 생산성 및 개발 코드의 재사용성을 높일 수 있도록 하였다.

3.2 웹 기반 개발 환경에서의 적용

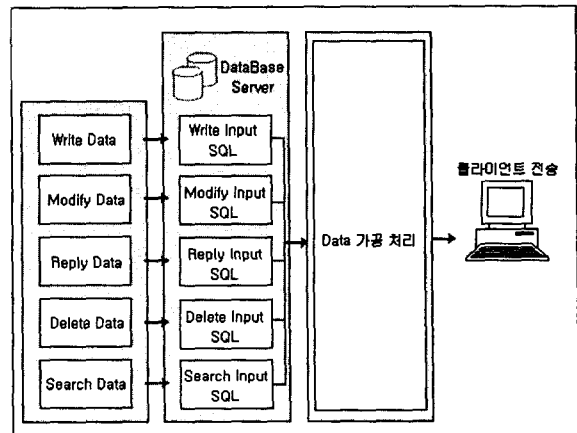
MVC 기반의 웹 컴포넌트 컨테이너 시스템을 적용하기 위

해서 웹 어플리케이션에서 가장 많이 사용되는 게시판 개발에 적용하기로 하였다. 게시판의 개발 로직을 분석하게 되면 [그림 1]과 [그림 2]와 같이 나타낼 수 있으며, 게시판 자료의 흐름은 서버에서의 해석과 클라이언트 요청으로 분리될 수 있다.



[그림 1] 게시판 클라이언트 패턴 분석

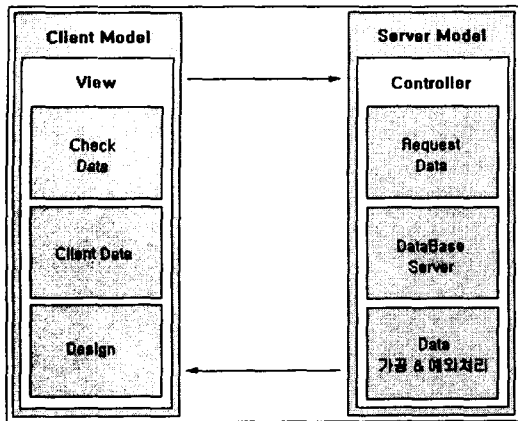
[그림 1]과 [그림 2]와 같이 각각의 패턴단위를 MVC 형태의 분리를 시키면 [그림 3]처럼 서버와 클라이언트로 모델 단위로 구분이 가능해진다. View 부분은 웹 형태의 디자인으로서 개발 단위 모델로 구현되며, 디자인 부분과 서버 전송 부분으로 나뉜다. 전송을 하기 전에는 항상 Check Data 부분을 컴포넌트화 하여 데이터의 유효성을 체크해야 한다. View부분에서 사용자 인터페이스 부분과 개발 코드 부분을 분리시킬 수 있다.



[그림 2] 게시판 서버 패턴 분석

서버 부분은 사용자가 요구하는 각각의 이벤트 부분을 단위 별로 최소의 패턴을 컴포넌트화하여 HTML내에 스크립트 형식으로 프로그래밍 되지 않도록 Controller부분의 단위를 만든다.

여러 처리의 경우 [그림3]처럼 Client Model의 경우 Check Data 처리를 하며, 서버 모델은 예외 처리 부분 따로 Data 가공 & 예외처리에서 처리되도록 한다.



[그림 3] MVC 응용 게시판 패턴

실제 구현시에는 .NET을 이용하였으며, 게시판의 패턴을 분석하여 프로그래밍이 되어 있는 모든 부분을 각각 컴포넌트화를 시켜서 \*.dll 파일로 등록을 시킨 후 각각의 컴포넌트를 합쳐서 Container에 등록하여 처리를 한다. 필요한 경우 해당 Container에서 필요한 부분을 사용자 인터페이스 부분과 매칭을 시킨다. 매칭을 시키는 경우 미리 정의해 놓은 예약된 코드와 해당 컴포넌트를 점목 시키도록 처리를 한다 해당 컴포넌트에 처리된 파일은 웹 서버를 통하여 클라이언트에게 전송 한다.

3.3 웹 기반 개발 환경에서의 적용 결과

MVC 기반의 웹 컴포넌트 컨테이너 시스템을 이용하여 웹 어플리케이션을 개발하는 경우 사용자 인터페이스와 개발 코드 부분은 자연스럽게 분리가 가능해지며, 사용자 인터페이스와 분리된 개발 코드는 다른 부분에 재사용이 가능하다. 또한, 디자인과 개발 코드가 영향을 미치지 않으므로 개발 생산성도 향상시킬 수 있다. [표 1]은 MVC 기반의 웹 컴포넌트 컨테이너 시스템을 적용하지 않은 개발 코드와 적용한 개발코드를 비교한 것이다.

4. 결론

컴포넌트 기반의 개발 방법은 소프트웨어 공학에서 활발하게 연구된 분야였으며 C/S환경에서 다양하게 적용되어져 왔다. 그러나, 웹 기반 개발 환경에서는 웹의 특성상 적용하기가 상당히 어려웠다. 그러나, 본 논문에서 제안한 MVC 기반 웹 컴포넌트 컨테이너 시스템을 활용하여 개발하는 경우 쉽게 적용할 수 있으며, 컴포넌트 환경의 장점인 코드의 재사용성 및 개발 생산성을 향상시킬 수 있다. 향후 연구과제로는 좀더 체계적으로 웹 관련 컴포넌트들을 개발하여 다양한 환경에서 쉽게 적용할 수 있도록 할 것이다.

개발 소스 부분	
적용 전	<pre> &lt;OBJECT RUNAT=server PROGID=ADODB.Connection id=oConn&gt;&lt;/object&gt; &lt;OBJECT RUNAT=server PROGID=ADODB.Recordset id=rs&gt;&lt;/object&gt; &lt;% oConn.open DSNName %&gt; &lt;% rs.Open "select * from board where ORDER BY num DESC", oConn, adOpenStatic %&gt; &lt;% for i = 1 to a_PageSize %&gt; &lt;%title = rs("title")%&gt; &lt;tr&gt; &lt;td width="26" height="25" class="td" bgcolor="#F1F1F1"&gt; &lt;%title%&gt; &lt;/td&gt; &lt;/tr&gt; </pre>
적용 후	<pre> //컴포넌트 부분(*.dll파일로 등록) public DataSet GetAllRespondent() { sqlConn = new SqlConnection(strConn); adapter = new SqlDataAdapter("select * from ITCboard Order by number desc",sqlConn); ds = new DataSet(); adapter.Fill(ds); return ds; } //컴포넌트 호출 부분 private void button1_Click(object sender, System.EventArgs { MyComponent.Data data = new MyComponent.Data(); DataSet ds = data.GetAllRespondent(); listBox1.DataSource = ds.Tables[0].DefaultView; listBox1.ValueMember = "title"; //디자인 부분 &lt;asp:ListBox id="ListBox1" runat="server"&gt;&lt;/asp:ListBox&gt; &lt;asp:Button id="Button1" runat="server" Text="Button"&gt;&lt;/asp:Button&gt; </pre>

[표 1] 개발 코드의 비교

5. 관련문헌

- [1] <http://my.netian.com/~oursky/webdev/>
- [2] 정강용, "웹프로젝트에서 디자인과 비즈니스 로직의 분리를 위한 ProDesigner시스템의 설계 및 구현", 순천대학교 박사논문, 2003.
- [3] <http://soomsori.net/moa/moin.cgi/ModelViewController>
- [4] [http://www.codeproject.com/csharp/Model\\_View\\_Controller.asp](http://www.codeproject.com/csharp/Model_View_Controller.asp)
- [5] <http://bizcom.kookmin.ac.kr:7001/community/front/qa/Q-List.jsp?boardkind=14&communityid=8>
- [6] <http://www.taeyo.pe.kr/lecture/NET/WebFormContorl01.asp>
- [7] 이경하,이규철 "웹서비스의 향후 발전 방향", 정보처리학회지 Vol.09 No.04(15), 2002.