

## .NET 스마트 웹 개발을 위한 매니즈드 하이브리드 웹 구현

장승영\*, 오정진\*\*, 정병수\*\*\*

### A Managed Hybrid Web implementation for a .NET Smart Web development.

Seung-Young Jang \*, Jeong-Jin Oh \*\*, Byeong-Soo Jung\*\*\*

#### 요약

본 논문은 하이브리드 웹에서 UI제작과 표현에 따른 문제점들을 해소시키기 위한 방법으로 매니즈드 하이브리드 웹을 적용하게 되었다. 매니즈드 하이브리드 웹의 적용은 .NET 스마트 웹 개발의 효율성을 높이는데 그 목적이 있으며, 구조화 시스템 설계 방식은 인터페이스 객체에 대한 효율성과 확장성을 높일 수 있게 되었다. 특히 클래스 구성방식에 있어서 CS 환경에서 가능한 구성 요소를 적용함으로써 객체의 효율성과 웹에 대한 접근을 개선시킬 수 있었다. 무엇보다 객체의 공존성과 소스의 편의성을 한층 더 높이게 되었다. 결론적으로 사용자 인터페이스 측면에서 UI에 대한 제약 없이 그 표현방식이 자유롭고 각 파트별 독립성을 유지하게 되었다. 이러한 적용과 개선은 웹만이 지니는 표현방식과 개발에 대한 자원의 효율성을 높일 수 있게 되었다.

▶ Keyword : .NET, 스마트 웹, 하이브리드 웹, UI, 클래스

#### Abstract

The purpose of this paper is to suggest a method that .NET smart web development is enhanced through Managed Hybrid Web to improve the problems of UI development and representation in Hybrid Web. Improvement of structured system design of Managed Hybrid Web can enhance the efficiency and extendability of the interface objects. In particular, Source convenience is further enhanced regarding to coexistence of the objects if a client and server technique is applied to the class generation ways. It comes to the conclusion that the application of the user interface has the

• 제1저자 : 장승영 • 교신저자 : 오정진

• 투고일 : 2011. 11. 14, 심사일 : 2011. 11. 23, 게재확정일 : 2011. 12. 14.

\* 남부대학교 디지털경영정보학과(Dept. of Digital Management Information, Nambu University)

\*\* 전남과학대학교 자치경찰과 교수(Dept. of Self-Government Police, Chunnam Techno College)

\*\*\* 남부대학교 디지털경영정보학과 교수(Dept. of Digital Mangement Information, Nambu University)

free representation ways, and the extension of representation ways can be achieved by each part independence.

▶ Keyword : .NET, Smart Web, Hybrid Web, UI, class

## I. 서 론

최근 정보통신산업진흥원에서 발표한 ‘2011 SW 기술 이슈 TOP 10’ 자료에 의하면 스마트는 중요한 SW(Software) 비즈니스 키워드로 영향을 미칠 것이라는 전망과 함께 발표를 내놓았다. 현재 스마트폰 시장은 급격한 성장과 스마트 관련한 비즈니스가 확산되어지고 있으며 그로 인한 SW 기업 및 관련 비즈니스에 큰 변화가 일어나고 있다. 무엇보다 이 보고서에 주목할 부분은 인터넷 웹과 모바일 애플리케이션을 결합한 형태인 하이브리드 웹(Hybrid-Web)의 등장이다. 스마트폰의 확산과 인터넷 웹의 모바일 환경에 대한 최적화 요구의 소리가 높아지고 이를 반영한 인터넷 웹과 모바일 애플리케이션이 결합된 형태인 하이브리드 웹(Hybrid-Web)이 새로운 이슈로 등장하게 되었다[1].

특히, 대기업들의 국내 모바일 광고 시장에 진출을 주목해 봐야 한다. KT경제경영연구소 따르면 모바일 광고 시장 규모가 2015년에 1조 1,000억 원 규모가 될 것으로 전망하고 있다[2]. 대기업들의 이러한 모바일 광고의 시장성 확보를 위해 결국 다양한 모바일 서비스를 소비자에게 제공해야만 했으며 이렇듯 다양한 콘텐츠 제공은 하이브리드 웹에 등장으로 다소 해결 요소를 찾게 되었다.

초기 스마트폰 시장은 Apple App Store를 중심으로 집중화 되어 있었으며 채널을 통한 에코시스템(Ecosystem)을 형성하여 모바일 앱을 사용자에게 제공해 왔다[3]. 그러나 사용자들은 PC상에 사용자 중심의 UI(User Choice Interface) 패턴방식이 모바일에서도 표현되길 원했다. 결국 접근성과 사용성을 고려한 최적의 특징을 표현하는 웹 방식으로 하이브리드 웹의 기술이 도입되어진 것이다. 하이브리드 웹을 한마디로 정의한다면 모바일 웹과 모바일 앱의 “기능융합”이라고 정의 할 수 있다. 하이브리드 웹은 배포, 디바이스(Device) 접근 제어, 로컬 스토리지(Local Storage) 접근, Push Notification 등의 장점을 가지고 있다. 또한 효율성을 제공하는 웹기반의 다양한 플랫폼을 지원가능 하게 되었다. 무엇보다 웹페이지 소스를 활용 가능한 호환성이 뛰어난 점에 가장 주목 되는 기술 부분이기도 하다[4].

결과적으로 이를 통한 PC상에서 구현 되어진 네이티브 웹

과 같은 효과를 모바일 웹에서도 구현할 수 있게 되었다. 하지만 이러한 차세대 모바일 하이브리드 웹에서도 구현방식과 설계방식에 대한 문제점이 대두되어지고 있으며 이러한 방식의 문제가 결국 다양한 플랫폼의 제공 한계점을 지니고 있다고 할 수 있다[5]. 이러한 구현방식의 한계점을 극복하기 위한 목적으로 본 연구에서 매니즈드 하이브리드 웹을 제안 하게 되었다.

이러한 제안은 매니즈드 하이브리드 웹의 기술적 적용 방식보다 개발 방법론적 측면에서 큰 비중을 두었으며 UX(User eXperience)와 UI(User Choice Interface)의 다양성 확보를 위한 표현 접근 기술방식에 다각적으로 제시하려고 하였다.

또한, 정적 콘텐츠의 하이브리드 웹에 동적 콘텐츠의 C#(C-sharp) 코드를 적용 하여 코딩(coding)의 재활용과 자원에 대한 효율을 높일 수 있게 되었다. 하이브리드 웹의 장점으로 부각되어지고 있는 접근성 부분에서도 한 단계 높게 개선시킬 수 있었다. 보다 쉽게 개발하고 보다 다양한 플랫폼을 사용하여 사용자 편의성 측면에서도 개선 된 개발방법론을 제시하고자 한다.

본 논문 구성은 II장에서는 관련연구로 하이브리드 웹의 주요 기술방식 등을 정의하였고, III장에서는 하이브리드 웹에 대한 기술적 결합방식의 도입에 대한 타당성을 제시하였으며, IV장에서는 이러한 기술적 결합방식을 적용하여 스마트 웹의 설계 패턴을 기능별 클래스 설계방식으로 구성하였으며, V장에서 기능적 클래스 설계 패턴으로 적용한 매니즈드 하이브리드 웹을 스마트 웹으로 구현하였으며, VI장의 결론을 통하여 매니즈드 하이브리드 웹의 구조화 시스템의 확장에 대한 방향을 설명하고 결론을 맺고, 향후 연구 방향에 대해 제시하였다.

## II. 관련연구

현재 하이브리드 웹은 차세대 스마트 웹으로 지칭하고 있다. 또한 모바일 웹의 지향점인 하이브리드 웹 진화로 스마트한 웹으로 성장 하고 있다[4].

이러한 성장은 기업의 모바일 광고 시장에서 찾아볼 수 있다. 과거 모바일 시장 광고를 위하여 기업들은 다양한 콘텐츠를 소비자에게 제공하려고 하였으나 다양한 스마트폰의 운영체제( Operating System)별로 고가 제작비용이 드는 모바일

앱을 투자한다는 관점에서 큰 부담감으로 작용 되어져 왔다. 그래서 기업들은 모바일 웹서비스를 제공 하려는 노력을 해왔다. 하지만 일반 홈페이지와 같은 다양한 웹 프로그램 사용과 스마트 앱의 주요기능에 대해 컨트롤 할 수가 없다는 문제점 때문에 소비자가 원하는 기능을 100% 수용할 수가 없었다. 그 대안으로 기능적 제한이 없는 하이브리드 웹의 적용 기술은 스마트 웹서비스를 제공하려는 많은 기업들에게 다양한 스마트 웹서비스를 제공할 수 있다는 사실에 많은 큰 관심을 불러일으키기 시작하였다[6].

하이브리드 웹 등장은 스마트 웹이 나아갈 방향을 제시하였으며 웹서비스의 표준화를 제안하게 되었다. 기업의 홍보나 마케팅(Marketing)에 이용되는 스마트워킹(Smart Working)의 기본적 업무 베이스(Base)로 활용할 수 있었기에 그 효용 가치가 매우 높아졌다[7].

하이브리드 웹의 사전적 의미는 스마트폰으로 도메인을 입력하여 접속하는 ‘스마트 웹사이트’와 애플리케이션을 다운 받아 접속하는 ‘앱(App)’이 합쳐진 형태로 정의를 내릴 수 있다. 하이브리드 웹의 원리는 웹 표준을 준수하여 모바일 웹을 만든 이후 앱에 연동하는 형태다. 따라서 애플리케이션을 다운 받거나 또는 북마크, 도메인을 입력해서 접속해도 동일한 형태의 모바일 페이지를 볼 수 있다는 장점을 지니고 있다. 이러한 하이브리드 웹은 차세대 웹 어플리케이션 표준화를 제시한 기술표준화 언어가 HTML5이라고 할 수 있다[1].

표 1. 모바일 응용의 유형별 기능 비교  
Table 1. A type facility comparison of a mobile application.

전략 범위	내장형 APP	웹 APP	하이브리드 APP
그래픽 성능	상	상	상
앱스토어 판매	가능	불가능	가능
오프라인 지원	가능	일부가능	가능
매시업	불가능	가능	가능
단말기 기능이용	용의	불가능	가능
SW업데이트	재설치	상용중수정	부분재설치
UI 제작·표현	상	하	중

HTML5는 차세대 웹 표준이며 단순 텍스트(Text)와 하이퍼링크(Hyperlink)만 표시하던 HTML이 복잡한 어플리케이션까지 제공할 수 있는 웹 어플리케이션 플랫폼으로 진화한 형태라고 말할 수 있다[8].

HTML5의 주요 특징을 간략하게 살펴보면 다음과 같다[1].

- 첫째, 우수한 2D/3D 그래픽과 동영상 제작 지원
- 둘째, 네트워크가 끊겨도 오프라인에서 애플리케이션을 계속 사용할 수 있도록 지원
- 셋째, 웹브라우저 상에서 LBS(Location Based Service)와 같은 어플리케이션을 구동
- 넷째, 웹서버와 효율적인 소켓통신으로 서비스 속도 향상 등이다.

이렇듯 HTML5의 핵심 내용을 요약 해보면 호환성, 유용성, 상호 운용성, 접근성을 원칙으로 하고 있음을 알 수 있다[2].

### III. 매니즈드 하이브리드 웹 (Managed Hybrid-Web) 기본 요소

매니즈드(Managed)란 세심히 관리된다는 뜻을 가지고 있으며, 매니즈드 하이브리드 웹에 사용언어는 매니즈드 언어에 속하는 C#(C-sharp)언어를 기본 언어로 적용하게 되었다.

그 이유는 C# 언어 자체가 모든 것을 객체로 취급하는 컴포넌트 프로그램 언어 역할을 수행하기 때문이다. C# 언어의 시작은 2000년 6월 마이크로소프트가 닷넷(.NET) 플랫폼을 위해 개발하게 되었으며, C++의 객체지향성과 자바의 분산 환경 처리에 적합한 다중성 등의 장점만을 가지고 있는 컴포넌트기반의 소프트웨어다. C#은 별도의 코드를 만들 필요가 없는 이식성이 높은 응용프로그램으로써 대표적인 매니즈드 언어에 속한다[10].

C#은 CRL(Common Language Runtime) 가상머신 위에서 작동하는 닷넷 프레임워크(.net Framework)를 사용한다. 또한, CRL은 공동언어기반 하에 캡슐화 된 가장 중요한 컴포넌트 언어 기반에 속한다. 과거 C#에서는 WAP(Wireless Application Protocol)을 개발환경에 적용하여 브라우저 기반으로 무선 인터넷을 제공해 왔었다. 또한, GVM(General Virtual Machine) 확장성을 제공하여 무선 단말기의 기능을 무한 확장 가능하게 했다. 그리고 API(Application Programming Interface)의 제공은 개발의 제약성이 없으며 COM(Component Object Model)을 쉽게 사용할 수 있게 되었다[3].

COM의 사용은 확장성에 큰 의미를 부여할 수 있는데 그 중에 하나가 여러 언어와 도구에서 범용적으로 사용할 수 있다는 점을 들 수 있다. 범용적 사용은 닷넷 프레임워크에서 쉽게 변환되어 구현할 수 있는 언어적 차원에서 가장 많은 지원을 한다. 이러한 구조 여건 때문에 PC상의 WinForms와

같은 UI 기능을 HTML 5 구조 하에서도 지원이 가능하게 되었다[5].

이렇듯 구조적 기능 지원 여부 때문에 하이브리드 웹의 UI 제작·표현에 대한 문제 해결과 스마트 웹 환경을 매니즈드 하이브리드 웹으로 구현 가능한 환경으로 제공할 수 있게 되었다. 언어차원 측면에서도 가비지 컬렉터(Garbage Collector)를 들 수 있는데 이는 객체를 검사하여 사용하지 않는 객체를 메모리에서 자동 제거해 주는 역할을 수행 한다. 이러한 제거 역할 기능 때문에 스마트 앱과 UI 기능 구현에 매우 유용한 도움을 주고 있다. 그 결과 웹에 정보제공과 서비스를 빠르게 교환하고 하나 이상의 운영체제에서 사용할 수 있는 패러다임을 반영 할 수 있게 되었다[8].

#### IV. 매니즈드 하이브리드 웹(Managed Hybrid-Web) 스마트 웹 구현

##### 1. 구현 환경

표 2. 구현환경  
Table 2. The implementation environment.

구분	사항
OS	Windows Server 2008 R2 Standard
웹 서버	IIS 7.0
Database	MS SQL 2008, oracle 9i
Database 연동	OLE DB
서비스 플랫폼	.net Framework 4.0
구현언어	C#, HTML5
개발 툴	Visual Studio 2010
테스트환경	아이폰4, Internet Explorer v9

시스템 구현을 위한 서버의 운영체제는 Windows Server 2008 R2 Standard 로 사용하여 운영하였다. 운영체제에서 기본으로 지원 되는 IIS 7.0과 서비스 플랫폼을 이용하여 서비스를 제공하였다. 데이터베이스는 MS SQL 2008의 사용과 oracle 9i를 OLE DB로 연동하였다. 마이크로소프트에서 개발 한 Visual Studio 2010 개발 툴을 사용하여 .net Framework 4.0 기반 하에 구현하였다. 스마트 웹 폼은 HTML5 언어를 사용하였고, 속성의 코딩은 C# 언어로 처리하여 화면에 제공되는 디자인과 보안 부분에 초점을 맞추었다.

##### 2. 클래스(Class) 설계

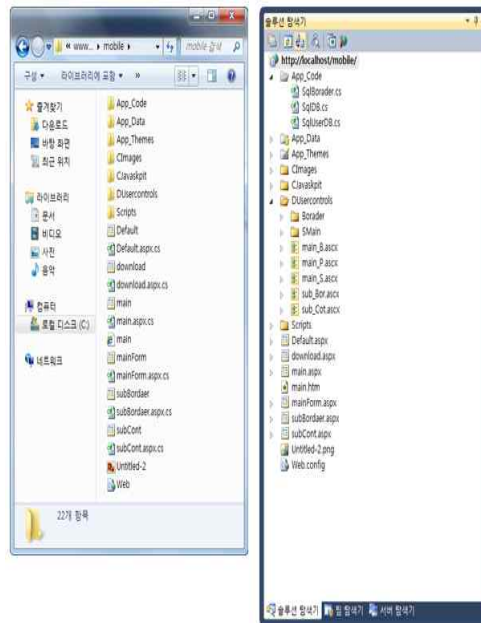


그림 1. 폴더 구조 및 프로젝트 구조  
Fig. 1. A folder structure and project structure

클래스(Class) 설계는 구조화 시스템(Organization) 설계 방식을 적용하여 설계하였다.

기본 클래스 설계는 역할부분으로 나누어 크게 클래스 구성과 인터페이스 구성의 공존성을 바탕에 두고 구성하였다. 이러한 설계 구성은 객체의 편의성과 효율성 부분에서 효과를 얻을 수 있다.

객체의 인터페이스 측면은 다중상속클래스와 추상클래스를 사용해서 구현하였다. 폼의 효율성 부분에서는 사용자 정의 컨트롤을 사용하여 구현하였다. 결과적으로 클래스 설계 적용 방식은 확장성을 높일 수 있었으며, 개발 편의성에서도 그 효율성을 갖게 되었다.

##### 3. 클래스(Class) 구성

클래스 구성은 각 역할에 쓰임새, 객체의 공존성에 따라 나누어 설계 하였다. 크게 기능 역할별로 3개로 나누어 객체간에 참조성 트리(Tree) 구조로 설계 방식을 적용하였다.

객체는 기본객체의 클래스와 구성객체의 클래스 간에 객체에 대한 인터페이스의 오류를 줄여 각 객체에 대한 구분 없이 사용할 수 있게 편의성을 고려하였다. 또한 객체의 편의성은 객체간의 인터페이스의 관리가 용이 하도록 설계하였다.

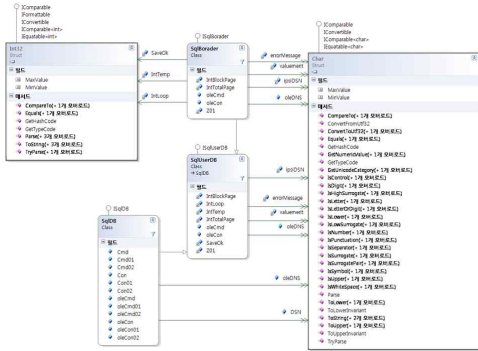


그림 2 클래스 다이어그램  
Fig. 2 A class diagram

메인 메소드인 엔트리 포인트(Entry Point)를 사용하여 접근에 제약을 두지 않아 객체 간에 인터페이스를 각각에 구현해야 하는 불편함을 줄일 수 있었다. 또한, 자료형의 값을 가져오는 형태로 구현해 주는 방식을 적용하여 구성 객체가 동일한 상속 구조로 구현하여 소스코드의 편의성을 적용하였다.

무엇보다 객체의 공유를 관리 적용하여 공유 되지 않는 객체들에 대한 소멸처리가 실행될 수 있도록 고려하여 구현하였다.

#### 4. 닷넷 프레임워크 클래스 구성

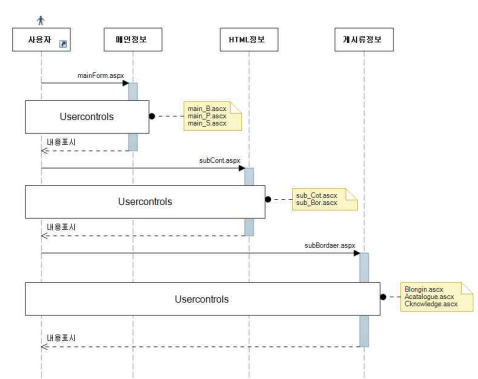


그림 3 유저컨트롤 다이어그램  
Fig. 3. An user control diagram

스마트 웹의 폼(Form) 구현은 사용자 인터페이스 화면 표시 부분과 로직(Logic) 두 부분으로 나누어 구성하였다. 정적으로 생성 되는 HTML5 전달방식과 C#코드로 작성된 동적 전달방식이 실행되어 하나의 표준 문서가 작성 되어 진다. 이러한 동적 전달방식은 .aspx 확장자를 가진 별개의 파일로 저장 되어 진다.

작성된 하나 이상의 표준 문서는 .aspx 확장자를 통해 사용자에게 제공하게 된다. 또한 .aspx는 UI 페이지와 코드가

각각의 동적 메소드를 호출하여 하나의 컨트롤로 표준문서로 저장 되어 진다. 이처럼 동적 전달방식 구성은 System.Web 와 System.Web.UI 네임스페이스에 포함된 라이브러리 (FCL)에 의해 풍부한 폼 타입을 제공하게 되었다.

결과적으로 UI를 담당하는 코드 부분이 캡슐화 되어져서 컨트롤 속성을 통해 HTML5의 페이지와 연동 하게 된다. 이후 연동된 이벤트는 내용을 서버에 즉시 전송 처리하여 사용자에게 제공하는 방식으로 구현하였다.

### V. 실험 및 결과

#### 1. 인터페이스 영역

매니즈드 하이브리드 웹 서비스 형태는 역할별 쓰임새에 따라 크게 2가지로 분류하여 구성하였다.

첫째, 인터페이스 영역

둘째, 정보를 텍스트기반으로 제공하는 HTML 영역이다.

모든 설계 구성 방향도 동일하게 클래스 설계로 적용하였으며, 인터페이스 간의 상속관계를 유지하여 개발과 적용에 편의성을 확보 할 수 있게 되었다.

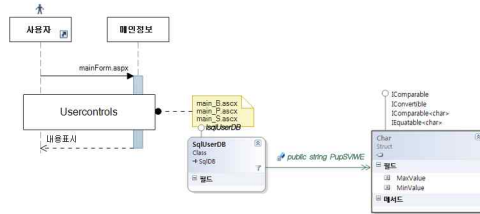


그림 4 메인화면 클래스 호출 흐름도  
Fig. 4. A main screen class call flow chart.

메인화면에서는 사용자 인터페이스 화면 구성을 콘텐츠 영역과 링크 영역으로 구분하여 적용하였다. 콘텐츠 영역은 DB 정보를 호출하며, 링크영역은 HTML 정보를 호출하여 각 기능에 맞는 정보를 제공하도록 구성하였다.



그림 5 메인화면 각 영역별 사용자 인터페이스 호출 (mainForm.aspx 파일)  
Fig. 5. A main screen each area user interface call. (mainForm.aspx File)

DB정보 호출은 영역별로 구성 된 클래스를 호출 DataBind 하게 된다.

```

DUsercontrols/main_S.ascx.cs*
protected void Page_Load
(object sender, EventArgs e)
{
    mf01 += "<marquee direction='up'
width=182 height='15'
scrollamount='1'
scrolldelay='100'
onmouseover='this.stop()';
onmouseout='this.start()';>";
    mf01 += UpCode.PupSVIWE
("mbbs001", "2a", "2");
//학교소식 호출
    mf01 += "<br>";
    mf01 += UpCode.PupSVIWE
("mbbs002", "2b", "2");
//학교공지사항 호출
    mf01 += "<br>";
    mf01 += UpCode.PupSVIWEipsi
("mbbs001", "1a", "2");
//입장정보 호출
    mf01 += "</marquee>";
    Page.DataBind();
}
    
```

그림 6. 메인화면 정보 클래스 호출 코드  
(main\_S.ascx.cs 파일)  
Fig. 6. A main a screen information class call code.  
(main\_S.ascx.cs File)

그리고 DB정보 클래스를 호출은 각 항목의 값을 정해 결과의 값을 추출 할 수 있도록 코딩 하였다. 이러한 값의 설정 방법으로 추출하여 클래스의 객체의 공존성을 고려하여 코딩 하였다.



그림 7. 메인 실행 화면  
Fig. 7. A main execution screen.

## 2. HTML 영역

HTML 영역은 텍스트기반으로 정보를 제공하는 영역으로 구성하였다.

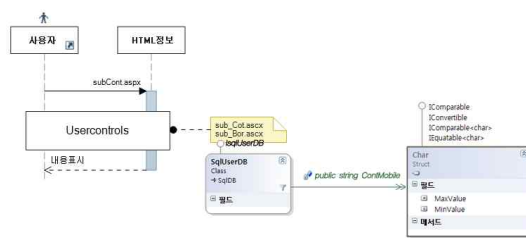


그림 8. 영역 클래스 호출 흐름도  
Fig. 8. An area class call flow chart.

메인화면 영역에서는 단일화된 영역의 화면 구성으로 정적인 사용자인터페이스를 호출 하였다. 반면 HTML 영역은 동적인 사용자 인터페이스를 호출하기 위해 접근의 용이성을 고려하였다. 그래서 컨트롤 클래스를 상속 받기 위해 Panel 컨트롤을 사용 하였다. Panel 컨트롤을 배치시켜 패턴의 값을 적용해 그 값에 따른 정보를 호출 할 수 있도록 하였다.

```

mainForm.aspx.cs*
protected void P_Conmore01_Load
(object sender, EventArgs e)
{
    if ((int.Parse(serve03.ToString().Trim()))
//값설정
    {
        P_Conmore01.Controls.Add(LoadControl
("~/mobile/Dusercontrols/sub_Cot.ascx")
//호출
    }
}
    
```

그림 9. HTML 영역 사용자인터페이스 호출 코드  
(mainForm.aspx.cs 파일)  
Fig. 9. HTML an area user interface call code.  
(mainForm.aspx.cs File)

텍스트기반으로 정보를 호출은 영역별로 구성 된 클래스를 호출 DataBind 하게 된다.

```

DUsercontrols/sub_Cot.ascx.cs*
protected void Page_Load(object sender, Ever
{
    if (serve03.ToString().Trim() != null )
    {
        Imgurl = "<img src='/mobile/CImages
/SubForm/c00 "
+
serve03.ToString().Trim()
+ ".png' />"
        contnaem = UpCode.ContMobile
(serve03.ToString().Trim());
        contday = "";
        Page.DataBind();
    }
}
    
```

그림 10. HTML 영역 클래스 호출 코드  
(sub\_Cot.ascx.cs 파일)  
Fig. 10. HTML an area class call code.  
(sub\_Cot.ascx.cs File)

그리고 텍스트기반으로 정보를 DB에서 컨트롤 될 수 설계 하여 각 항목을 클래스에 호출하여 값을 가지고 오게 된다. 이러한 DB에서 컨트롤 될 수 있게 설계하고 호출은 신속한 정보제공을 고려한 사항이다.

```

App_Code/SqlUserDB.cs*
hstorCK += "<table width='310px' cellspac
+\"ing='0' cellpadding='0' border='0
if ( reader.Read() )
{
hstorCK += "<tr><td width='10'></td>
hstorCK += "<tr>";
hstorCK += "<td width='300' valign='top'
+\"align='left'>";
hstorCK += "<span style='font-size:9pt;'
+\"lang='EN-US'>";
hstorCK += reader["cmsmode"].ToString();
hstorCK += "</span>";
hstorCK += "</td>";
hstorCK += "</tr>";
}
hstorCK += "</table>";
hstorCK += "<br/><br/>";

reader.Close();
Con.Close();

return hstorCK;
}
    
```

그림 11. HTML 영역 DB 정보 클래스 코드 (SqlUserDB.cs 파일)  
 Fig. 11. HTML area DB an information class code. (SqlUserDB.cs File)



그림 12. 파트별 HTML 실행화면  
 Fig. 12. A part HTML execution screen

## VI. 결론

본 연구 목적은 하이브리드 웹에 대한 문제점을 극복을 위한 대안으로 매니즈드 하이브리드 웹을 적용 하게 되었다. 보다 빠르고 정확한 정보를 무선 네트워크 속에서 최강의 스피드로 대중화를 이루고 있으며, 스마트폰 시장성의 확대는

기업의 모바일 디바이스(Device)를 활용한 다양한 광고 마케팅 형태로 자리 매김 하게 되었다.

이처럼 소비자들의 특성에 맞춘 일대일 타겟 광고는 스마트 웹의 표준화의 접근성을 가져오게 되는 일등공신의 역할을 하게 되었다. 웹 표준화된 HTML5 언어의 확장성은 인식과 운용과 이해의 용이성에 대한 기술적 진보를 얻게 되었으며, 그 결과 스마트 웹의 새로운 기술인 하이브리드 웹이 관심을 갖게 되었다. 하지만 이러한 기술 표현방식에서도 사용자 중심의 다양한 콘텐츠를 서비스하기 까지는 다소 부족한 부분이 있다.

이런 관점에서 매니즈드 하이브리드 웹의 구현은 하이브리드 웹의 문제를 해결 하고 스마트 웹의 장점과 웹의 장점을 보다 효과적으로 사용자 중심의 표현방식을 제공 할 수 있게 되었다. 매니즈드 하이브리드 웹의 구조화 시스템(Organization) 설계방식의 적용은 인터페이스 객체에 대한 효율성 뿐만 아니라 확장성을 높일 수 있게 되었다. 이처럼 적용된 구조화 시스템 설계 방식의 분석절차는 단순한 시스템 구성 방식이 아닌 분야별 중심 과제로 분석이 순차적으로 이루어져야 하며, 이를 기초로 세분화된 설계방식이 아래와 같이 분석 되어져야 한다.

첫째, 자기 다른 운영체제를 고려한 모바일 웹 로드맵 (Road map)에 따른 역할 분석

둘째, 사이트 기본 중심 설계의 검색엔진에 대한 최적화 분석

셋째, 실시간 양방향성에 대한 인터페이스 측면 분석

넷째, 방문목적의 인지 높은 디자인요소 분석

다섯째, 화면영역에 대한 기본적 페이지 분석

여섯째, 정확한 내비게이션의 흐름 분석

일곱째, 콘텐츠 영역에 대한 영역별 분석

이러한 세분화된 분석을 통한 구조화 시스템 설계방식은 보다 쉽게 접근이 용이한 클래스 구성과 사용자 인터페이스 구성으로 개발의 편의적 측면에서도 성과를 얻게 되었다.

무엇보다 단순한 편의적 개발과정이라기 보다는 클래스 구성의 객체의 공존성에 의해 소스코드의 편의성을 더 효과적으로 높일 수 있게 되었으며, 사용자 인터페이스 구성인 UI 측면에서 제약성 없는 다각적 표현 방식의 확장성에 대해 검증할 수 있었다.

하지만 이러한 웹 표현기술 방식만으로는 모든 단말기의 기능 측면에서 완벽하게 구현하고 표현하는 되는 많은 문제점 지니고 있다. 현재 상용화 되고 있는 3D 그래픽 표준 기술이나 2D가속을 지원하는 환경에 맞는 구현 또한 어려움을 안고 있다. 음성, 영상, 멀티미디어 지원, 카메라 장치 제어 기능을

요구하는 앱에서 까지도 구현하기는 어려움이 있는 것이 현실이다. 이는 차세대 웹 표준이 개발 진행 중이며 각각의 단말기의 제어에 관한 표준 웹 기술에 대한 정의가 진행 중이기 때문이기도 하다. 따라서 웹과 웹기반 앱 개발 기술이 부족해서라기보다는 관련 웹 표준과 이를 지원하는 브라우저와 단말기 성능상의 한계 때문에 시간이 더 필요로 한 상황이다. 또한 웹과 웹기반 앱 개발 기술의 확장성에 관심과 표현방식의 연구가 계속 진행 되어져야 할 과제를 안고 있다.

### 참고문헌

[1] SW Insight Policy Report-2010-2011, "SW Issue Analysis and Forecast", National IT Industry Promotion Agency, 2011

[2] YTN News, "NHN, Advertising Sales 1,100,000,000,000 won. Full Advertising Market 14%", YTN [Economy], 2011.

[3] JungHoon Kim, MinSeok Jang, "Design and Implementation of Mobile Reservation System on Dot Net Framework", Proceedings of KFIS Spring Conference, Vol. 14, No. 1, pp. 543-546, 2004.

[4] S.Y. Lee, H.W. Jung, "Trends of Standardization on Future Mobile Web Platform", Electronics and Telecommunication, Vol.25, No. 3, pp.11-17, June. 2010.

[5] Won Seok Lee, "HTML5 and Mobile web", TTA Journal, No.128, pp. 50-54, 2010.

[6] EunMin Lee, "HTML5 in a Web environment impact", Korea Information Society Development Institute, Vol. 23, No 5-504, pp.54-65, March. 2011.

[7] GoEun Lee, JongWoo Lee, "Analyzing Effectiveness of the Web Application Development in Smartphone", Journal of Digital Contents Society, Vol. 11, No. 3, pp. 323-330, Sep. 2010.

[8] D.H. Kim, C. Ryu, J.H. Lee, S.J. Kim, "Mobile Software Platform Trends for Smartphone", Electronics and Telecommunication, Vol. 25, No. 3, pp.1-10, June. 2010.

[9] H.W. Jung, S.Y. Lee, "Standardization for Future Mobile Web Application", Electronics and Telecommunication, Vol.25, No. 1, pp.100-113, February. 2010.

[10] Jesse Liberty, "C # Programming", Hanbit Media, Inc 2004.

### 저자 소개



#### 장승영

2009 : 남부대학교 디지털경영  
정보학과 공학석사.  
2011 : 남부대학교 디지털경영  
정보학과 박사과정중  
현재 : 전남과학대학교 정보전  
산원 대리 재직 중  
관심분야 : Web 개발  
Email : sychaing@hotmail.com



#### 오정진

1996 : 호남대학교 컴퓨터공학과  
공학석사.  
2009 : 목포대학교 멀티미디어  
공학과 공학박사  
현재 : 전남과학대학교 자치경  
찰과 교수 정보보호연  
구소소장  
관심분야 : 컴퓨터공학  
Email : jjoh@chunnam-c.ac.kr



#### 정병수

1988 : 전남대학교 전산통계학  
이학석사  
2005 : 전남대학교 컴퓨터공학과  
이학박사  
1991~1999 : 전남과학대학 교수  
현재 : 남부대학교 디지털경영  
정보학과 교수  
관심분야 : 소프트웨어공학, 디  
지탈영상처리, 임베  
디드프로그래밍분야  
Email : bsjung@nambu.ac.kr