

스마트폰 상에서의 웹 응용프로그램 개발 환경 비교

Comparison Study of Web Application Development Environments in Smartphone

이고은, 이종우
숙명여자대학교 멀티미디어학과

Goeun Lee(goen0906@naver.com), Jongwoo Lee(bigrain@sm.ac.kr)

요약

기존 스마트폰 응용프로그램 개발 시 단점으로 부각된 다양한 플랫폼 환경 설정 고려와 이중 기기의 호환성 문제점 등 한계성이 늘어나면서 자연스럽게 모바일 웹 응용프로그램 개발이 대안으로 부각되고 있다. 모바일 웹 응용프로그램 종류의 하나인 하이브리드용 웹 응용프로그램은 스마트폰에 내장된 웹킷 엔진을 이용하기 때문에 간단하게 개발될 수 있다는 장점이 있다. 스마트폰의 웹킷 탑재로 응용프로그램 개발자는 HTML과 자바스크립트, CSS 만으로도 쉽고 빠르게 개발할 수 있을 뿐만 아니라 다양한 모바일 기기에 서비스를 제공할 수 있다. 웹킷은 스마트폰 웹 브라우저에서 모바일 친화적인 고성능 렌더링 엔진이다. 본 논문에서는 웹 응용프로그램 개발 시 고려해야할 웹킷 인터페이스의 성능과 기능을 비교 분석하였다. 또한 실제 모바일 웹 응용프로그램에 웹킷 메소드들이 어떻게 쓰이고 있는지 분석하였으며, 아이폰과 안드로이드폰 중 웹 응용프로그램 개발 시에 어떠한 환경이 개발자에게 편리한지도 분석하였다. 그 결과 아이폰 웹킷 성능이 안드로이드 보다 뛰어나다는 것을 알 수 있었다.

■ **중심어** : | 스마트폰 | 웹 응용 | 웹킷 | 개발 환경 |

Abstract

Due to the complex registration & downloading process of the native applications, and, software and their non-standardized APIs, mobile web application is now being an alternative software for smartphones. Hybrid web application, one of the types of mobile software, because develop and has reasonable performance by using the webkit engine in smartphones. It can be easily developed by using the exiting programming skill such as HTML, JavaScript and CSS. Additionally these programming techniques can be easily used in any smartphone regardless of its platforms. Most smartphones have a webkit engine or web rendering engine for high performance and smooth display in web browser. Webkit is now equipped in iPhone and Android phone. In this paper, we try to find out by comparison that the various aspects of webkit APIs of iPhone & Android phone, such as screen font size, screen orientation, touch event, gesture event and their performance. We also evaluated which one is more convenient for developers when making web programs using webkit. As a result, we found out that webkit in iPhone has more excellent performance than Android.

■ **keyword** : | Smartphone | Web Application | Webkit | Development Environments |

* 본 연구는 숙명여자대학교 2009년 교내연구비 지원에 의해 수행되었음

접수번호 : #101012-001

접수일자 : 2010년 10월 12일

심사완료일 : 2010년 12월 08일

교신저자 : 이종우, e-mail : bigrain@sm.ac.kr

I. 서론

스마트폰은 일반적으로 PDA, 모바일 PC 등에서 제공되던 다양한 응용 프로그램(이하 응용) 및 인터넷 기능과 이동전화 단말기의 고유기능을 결합한 휴대용 기기를 지칭한다[1]. 특히 스마트폰은 인터넷을 본격적으로 활용하는 모바일 인터넷 기기라는 점에서 기존의 휴대폰과 차별된다. 미국 시장조사기관 NPD 그룹의 2008년 11월 발표에 의하면 \$200정도 되는 3G 아이폰이 690만대 팔리면서 시장판매율 1위로 선정된 것처럼 아이폰은 스마트폰 시장에 큰 열풍을 일으키고 있다[2]. 아이폰과 경쟁하고 있는 안드로이드폰도 폴 브라우저 서비스를 제공하고 있다. 이러한 서비스는 스마트폰 웹 브라우저에 탑재된 웹킷(webkit)엔진에 의해 가능해졌다. 웹킷은 안정적인 인터넷 연결과 함께 뛰어난 모바일 웹 환경을 제공하는 웹 응용 개발용 API이다. 또한 스마트폰 선택에 있어서 가장 중요한 요인은 소비자가 만족할 만한 다양한 응용 프로그램 서비스의 제공 여부이다. 앱스토어는 2008년 7월 출시되어 각종 엔터테인먼트, 유틸리티, 교육, 게임 등의 다양한 응용을 제공하고 있다[4]. 2010년 1월까지 61억 달러의 매출을 낸 애플사의 선전에 따라 다양한 업체들이 모바일 응용 경쟁에 본격적으로 뛰어들고 있다. 이는 기존 이동통신사 중심의 폐쇄적인 모바일 응용 개발 및 유통구조가 마켓플레이스를 중심으로 개방적인 환경으로 변화하고 있음을 의미한다. 특히 아이폰과 안드로이드 플랫폼용 사용자 개발 응용의 인기가 높아지면서 모바일 웹 응용 분야에 새로운 기회가 다가오고 있다.

본 논문에서는 웹킷이 아이폰과 안드로이드에도 탑재되었다는 점을 활용하여 안드로이드와 아이폰 환경에서 웹킷 기능과 성능을 비교 분석한다. 이를 위해 안드로이드와 아이폰 개발 환경을 분석하고 웹킷의 구성 요소 및 렌더링 방식에 대해 살펴본 뒤 이를 통해 개발자에게 유리한 환경이 어느 것 인지를 제시하고자 한다.

II. 관련연구

일반 휴대폰과 스마트폰을 구별하는 가장 큰 요소는

일반적으로 범용 운영체제의 탑재 여부이다. 범용 운영체제가 탑재되어 있다는 것은 일반적인 컴퓨팅 환경에서처럼 다양한 응용 소프트웨어를 실행하기 위한 모든 시스템 서비스가 제공된다는 의미이고, 아울러 일반적인 컴퓨팅 환경에서 응용 소프트웨어를 개발, 구현하는 것과 유사한 편의성을 제공할 수 있다는 의미이다. 특히 애플사의 아이폰은 기존의 온라인 서비스를 기반으로 하는 새로운 모바일 응용의 개념을 정립하였으며 일반 소비자가 만드는 응용 소프트웨어를 유통할 수 있는 온라인 공간인 앱스토어를 제공함으로써 소비자의 참여에 의한 모바일 플랫폼 진화를 유도하였다. 다음은 안드로이드와 아이폰 플랫폼의 정의 및 구성요소를 알아보고 모바일 플랫폼의 진화로 일반 웹 기반 서비스를 그대로 혹은 모바일 기기에 최적화된 형태로 활용할 수 있는 모바일 브라우저를 분석한다.

2.1 안드로이드 모바일 플랫폼

안드로이드는 구글을 포함한 34개 글로벌 파트너로 구성된 OHA(Open Handset Alliance)[5]가 개발 중인 모바일 단말기에 탑재되는 개방형 플랫폼이다. 안드로이드 플랫폼은 모바일 환경에서 개발자들이 자유롭게 애플리케이션을 개발하여 탑재할 수 있도록 하는 통합적 SW 스택으로 OS와 API, 미들웨어 사용자 인터페이스, 브라우저에 이르는 모바일 기기에 들어가는 전반적인 SW환경을 지원한다. 안드로이드 SDK(Software Development Kit)는 개발에 필요한 기본적인 도구들을 포함하고 있으며, API를 제공하여 Java언어를 이용한 안드로이드용 응용프로그램 개발을 돕는다. [표 1]은 안드로이드 플랫폼의 주요 특징을 요약하고 있다.

표 1. 안드로이드 플랫폼 주요 특징

종류	설명
응용 프레임워크	컴포넌트를 교체하고 재사용할 수 있음
Dalvik VM	모바일 디바이스최적화
통합브라우저	webkit 기반
최적화된 graphics	2D 라이브러리와 Open GL ES 1.0 라이브러리 제공
Media 지원	이미지, 오디오, 비디오(MPEG4, H.264, MP3, AAC, AMR, JPG, PNG)

SQ Lite	데이터베이스
GMS telephony Bluetooth, EDGE, 3GWiFi camera, GPS,	하드웨어 의존적
풍부한 개발환경	에뮬레이터, 디버거, 프로파일러, Eclipse 플러그인

2.2 아이폰 모바일 플랫폼

애플사의 아이폰은 MAC OS X 버전 10.4.10을 사용한다. 이 플랫폼은 FreeBSD[6]기반의 Darwin[7]을 바탕으로 풍부한 멀티미디어 기능과 그래픽스 기능의 UI를 내세우고 있다. FreeBSD는 Intel 칩에서 실행되는 매우 빠른 UNIX 계열의 무료 OS이다. MAC OS X의 코어 부분은 Darwin이라고 불리는데, Darwin은 애플의 오픈 소스 라이선스 하에 공개 되어 있다. 2010년 애플에서는 아이폰에서 동작하는 소프트웨어 개발도구 버전4.0까지 발표했다. 아이폰 SDK 안에 포함된 Xcode[8]라는 개발 툴에서 코코아터치(Cocoa Touch)[9] 프레임워크와 Objective-C 언어를 사용하여 아이폰 응용을 개발할 수 있다.

아이폰 소스 코드는 보통 Objective-C 2.0으로 작성한다. Objective-C 2.0은 작고 강력한 ANSI C[10] 확장언어 세트로 설계된 객체 지향 구문으로 동적이며 유연성이 높다. 애플은 이 언어에 익숙하지 않은 개발자를 위해 아이폰 개발자 사이트[11]를 통해 Objective-C 레퍼런스 등 다양한 온라인 강좌를 제공하고 있다.

2.3 모바일 브라우저

아이폰이 스마트폰 시장점유율 1위를 기록하게 된 배경은 Wi-Fi나 무선인터넷을 통해서 내장된 사파리 모바일 웹브라우저[12]로 PC급 풀브라우저가 실현되었기 때문이다. 아이폰 열풍에 맞서 안드로이드 스마트폰 G1[13]이 2007년 개발되었다. 안드로이드폰의 장점은 구글의 서비스를 PC처럼 거의 그대로 사용할 수 있다는 점이다. Gmail이나 구글 캘린더, 구글맵스와 구글어스 등 다양한 구글웨어들을 PC에서 사용하듯 안드로이드폰에서 사용할 수 있다. 이러한 서비스는 구글에서 만든 웹 브라우저인 크롬(chrome)의 모바일 버전[14]이

탑재되어 있기 때문에 가능하게 되었다. 또한 크롬은 사파리와 동일한 웹킷 엔진을 사용하고 있어서 아이폰과 비슷한 수준의 풀브라우저가 가능하다. 웹킷은 안정적인 인터넷 연결과 함께 뛰어난 모바일 웹 환경을 제공하는 웹 응용 개발용 API이다. 아이폰이 매우 짧은 시간에 세계 제일의 모바일 웹 클라이언트로 성장한 것도 웹킷 엔진을 아이폰 플랫폼에 탑재했기 때문이다. 다음에는 웹킷에 대한 전반적인 구조를 알아본다.

2.4 웹킷

웹킷은 KDE(K Desktop Environment)의 KHTML[15]소스를 기반으로 애플, 노키아, 어도비, 구글 등에 의해 공동개발 중인 오픈소스 기반의 브라우저 응용 프로그래밍 엔진이다. 웹킷의 가장 큰 장점은 AJAX[16]를 지원한다는 점이다. 기존의 웹 응용은 브라우저에서 데이터를 웹 서버에 요청하고 되돌려 받았을 때 중복되는 값을 다시 한 번 전송 받음에 따라 많은 대역폭의 낭비를 유발한다. 반면에 AJAX 애플리케이션은 필요한 데이터만을 웹 서버에 요청하여 받은 후 클라이언트에서 데이터에 대한 캐싱(caching)처리를 할 수 있어서 웹 서버의 부하를 줄여 줄 뿐 아니라 처리 속도도 빠르다. 또한 HTML 4.01, CSS 2.0, JavaScript 1.5를 충실히 지원할 뿐 아니라 XMLHttpRequest 객체도 지원한다. 아울러 새로 정립되고 있는 최신 표준인 HTML5[17]도 지원 가능하다. [그림 1]은 웹킷의 구성요소를 보이고 있다. WebCore는 HTML 페이지 해석 및 렌더링을 담당하는 핵심이고, JavaScript Core는 자바스크립트 해석을 담당한다.

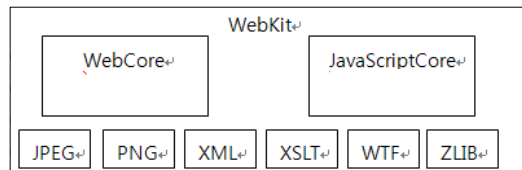


그림 1. 웹킷 구성요소

2.4.1 웹킷 렌더링

웹킷은 메모리를 효율적으로 사용하고 디바이스에

임베디드 하기도 좋으며 전체적으로 간결성을 유지하고 있어 새로운 브라우저 개발에서 베이스 코드로 사용하기에 적합하다. 웹킷 렌더링 클래스는 C+RTTI (RunTime Type Information)기법을 사용하여 오브젝트를 그린다. RTTI란 C++ 컴파일러 내에 포함되어 있는 기능으로써, 오브젝트의 유형을 실행 시에 결정할 수 있도록 허락하는 방식이다. 이 방식을 이용하면 프로그램 개발 시 사용하는 코드의 양이 많이 줄어들고 간단해지는 장점이 있다. [그림 2]는 웹킷 렌더링 클래스 구조를 보여주고 있다. 전체 구조는 웹뷰(WebView)라는 최상위 클래스가 있다. 이 클래스는 웹 전체를 제어하는 클래스이다. 하위 클래스로 Frame은 웹의 페이지 내부 레이아웃을 분할하는 역할을 하는데 여러 FrameView로 구성되어 있다. RenderObject는 각 오브젝트인 텍스트, 이미지, 테이블 등의 베이스 클래스로 공통의 인터페이스를 정해놓은 클래스이다. 웹 화면에 보여줄 문서나 이미지 정보가 도착하면 RenderObject를 상속받아 화면에 최적화 형태로 크기를 변경하여 최종 웹 콘텐츠를 그리게 된다.

다음 장에서는 각 스마트폰 환경에서 웹킷이 지원하는 기능에 대해 비교한다.

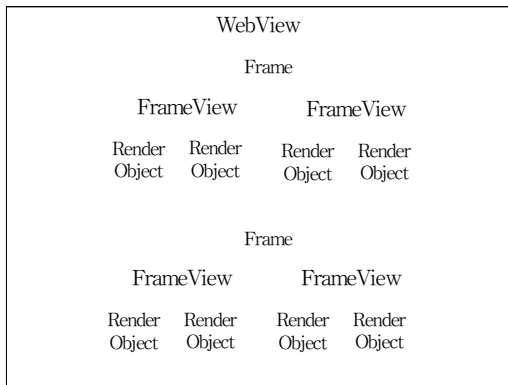


그림 2. 웹킷 렌더링 구조

III. 안드로이드와 아이폰의 웹킷 기능 비교

웹킷은 일반적인 웹 페이지를 모바일 친화적인 웹 페

이지로 원활하게 전환시키는 역할을 한다. 애플에서는 아이폰, 아이팟 터치에 사파리라는 이름으로 웹킷 기반 브라우저를 탑재 했으며, 구글은 안드로이드폰에 모바 일용 크롬을 탑재했다. 두 브라우저는 웹킷 렌더링 엔진을 기반으로 만들었다는 공통점이 있지만, 각 스마트폰 특성에 맞게 다양한 영역에 대한 수정 및 추가 등의 과정을 통해서 만들어 졌다. 이번 절에서는 웹 응용 프로그램을 개발 시 고려해야 할 안드로이드와 아이폰 인터페이스를 화면 크기와 화면 기울기, 터치 이벤트, 그리고 제스처 기능을 중심으로 비교 분석한다. 웹킷에는 이 4가지 말고도 훨씬 많은 API군이 있으나 본 논문에서 이 4가지를 중심으로 비교한 이유는 나머지 API들은 거의 차이가 없을 뿐만 아니라 일반 PC에서도 지원되는 항목이기 때문이다. 두 스마트폰 간 차이를 보이는 대표적인 이 4가지 API들을 중심으로 비교하도록 한다.

3.1 화면 크기

웹킷의 렌더링 엔진은 일반 사이트를 표시할 때 전체 페이지가 표시될 수 있도록 하기 위해 모든 콘텐츠를 매우 작고 세밀한 비율로 축소한다. 이때 개발자가 안드로이드폰이나 아이폰용 응용 개발 시 [표 2]와 같은 메타태그 viewport[18]를 이용하면 사용자가 효과적으로 콘텐츠를 볼 수 있다. 메타 태그 외에 추가적으로 아이폰에서는 개발코드에 CSS를 적용할 수 있어 안드로이드에 비해 내용에 독립적인 스타일 페이지를 구성할 수 있다는 장점이 있다.

표 2. 화면크기 사용 형식

	사용 형식
공통 메타태그	<metaname="viewport" content="width=device-width" />
아이폰 CSS	<linkrel="stylesheet"href="smallscreen.css" type="text/css" media="only screen and (max-device-width: 480px)" />
안드로이드	추가 기능 없음

3.2 화면 기울기

사용자는 웹 페이지 또는 응용 프로그램 실행 시 기

율기에 따라 세로형과 가로형의 두 가지 방법으로 전환해 가며 볼 수 있다. 아이폰 브라우저에서는 자바스크립트 window.orientation 속성을 활용하면 현재의 화면 방향을 알아낼 수 있다. 안드로이드폰에서 기울기 전환은 AndroidManifest.xml 파일에 사용형식 코드를 넣고 본문에는 해당 이벤트 핸들러를 넣어야 가능해진다. [표 3]은 아이폰과 안드로이드에서의 화면 기울기 방법을 요약하고 있다[19].

표 3. 화면 기울기 조절 사용 방법

	사용 형식
아이폰 자바스크립트	function setOrientation() { 변수=Math.abs(window.orientation) == 90 ?'landscape' : 'portrait' }
안드로이드	AndroidManifest.xml 추가 <android:configChanges="orientation keyboardhidden"> public void onConfigurationChanged() { if(화면 기울기가 세로 일 때) { 가로로 변환 } else {세로로 변환} } }

3.3 터치 이벤트

아이폰에서는 일반 PC에서 웹 페이지 선택 시 사용하는 마우스 이벤트 대신에 화면 터치 이벤트를 사용한다. 화면을 터치하면 아이폰 UITouch 클래스의 터치관련 함수들이 호출된다. UITouch 클래스는 어떻게 손가락이 화면 위에서 움직이는지 정의한다. [표 4]는 두 스마트폰 상에서 터치를 이용한 드래그와 드롭을 사용하기 위한 간단한 수도 코드를 보이고 있다.

표 4. 터치 이벤트 사용 방법

	사용 방법
아이폰	window.addEventListener(이벤트 변수('touchmove', function(event))) { 좌표 변수= event.targetTouches()}
안드로이드	public boolean onTouchEvent(MotionEvent event) { touchX = (int) event.getX(); touchY = (int) event.getY(); touchAction = event.getAction(); }

안드로이드폰에서 터치 이벤트를 실행 시키려면 [표 4]처럼 안드로이드 WebView 클래스에서 지원하는

onTouchEvent()를 응용 프로그램 소스 코드에 추가하면 된다. 안드로이드폰 화면을 누르면 터치 포인트의 위치를 onTouchEvent()에서 받는다. 이 함수에서는 현재 이미지 위치와 터치 포인트 간의 위치를 참조하여 움직여야 할 각도를 계산하고 픽셀단위로 이미지를 이동시킨다. 또한, 안드로이드폰에 있는 트랙볼도 onTrackballEvent()를 이용하여 사용할 수 있다[20].

지금까지 살펴봤듯이 터치 이벤트 처리에 있어서는 아이폰과 안드로이드폰 웹킷 API 사이에는 안드로이드에 트랙볼 처리 기능이 더 있다는 것 말고는 거의 기능 차이가 없다는 것을 알 수 있다.

3.4 제스처 기능

안드로이드폰에는 제스처 입력 기능이 안드로이드 2.0 버전부터 추가되었다. 2.0 이전버전에서는 제스처 입력 기능이 없는 반면, 아이폰에 탑재된 웹킷은 안드로이드와 달리 첫 버전부터 제스처를 지원한다. 제스처 기능에는 멀티 터치 인터페이스도 포함되어 있다. 멀티 터치 인터페이스를 통해 응용 프로그램에서 여러 손가락으로 디바이스를 터치할 때 발생하는 이벤트를 파악하고 처리할 수 있다. 제스처를 통해 할 수 있는 일은 스케일링(줌-인, 줌-아웃)과 회전이다. 줌(zoom)과 팬(pan)과 같은 제스처를 다루기 위해서는 제스처 이벤트 처리 코드인 gesturechange()를 이용하면 된다. [표 5]는 gesturechange() 이벤트 코드이다. gesturechange()는 멀티터치 이벤트에 반응하여 webkitTransform()을 호출한다. webkitTransform()은 scale 값과 rotate 값을 이용하여 스케일링과 회전을 할 수 있다[21].

표 5. 제스처 기능 사용 형식

	사용 형식
아이폰	window.addEventListener('load', function() { 이벤트 발생 변수('gesturechange', function(event) { 이벤트변수.webkitTransform=scale('값')+ rotate (' 값 ') false; }, false)
안드로이드	이벤트변수=newGestureDetector(this,new SimpleGestureListener()); 이벤트변수.onTouchEvent(event); public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY){}

안드로이드 제스처 기능은 안드로이드 자체에서 제공되는 서비스이므로 메니페스트는 수정하지 않는다. 왼쪽, 오른쪽, 아래, 위 네 가지 방향의 swipe(휘두르기) 동작은 거리 측정과, 속도를 통해 이루어진다. 제스처 인식의 핵심은 GestureDetector에 있다. 액티비티에서 OnGestureListener를 상속받으면 GestureDetector를 사용할 수 있다. 제스처 이벤트에는 OnDoubleTap, onDown, onLongPress, onScroll, onShowPress, onSingleTapUp, onFling 등이 있다. 이벤트 이름에서 알 수 있듯이 onDoubleTap은 두 번 터치하면 발생하는 이벤트이고, onDown은 터치하려고 손을 대기만 해도 발생하는 모든 제스처의 시작 이벤트이다. onFling은 onScroll에서 끝을 살짝 튕기는 동작에서 발생하는 이벤트이며, onScroll은 단어 그대로 스크롤 시 발생하는 이벤트이다. 안드로이드 제스처 인식은 세밀한 동작 하나하나 구현할 수 있다는 장점이 있지만 제스처 이벤트의 종료 시점을 알 수 없다는 단점이 있다[22].

3.5 기능 비교 결과

앞서 웹킷 기능 비교를 토대로 정리하면 다음과 같다.

- 화면크기: CSS 태그 사용 가능으로 인해 아이폰에서 좀 더 콘텐츠 성격에 의존적이지 않은 페이지를 만들 수 있다.
- 화면 기술기: 안드로이드에서는 AndroidManifest.xml 파일과 이벤트 핸들러를 모두 처리해주어야 하므로 번거롭다.
- 터치 이벤트: 차이가 거의 없음
- 제스처 기능: 안드로이드에서 좀 더 세세한 제어를 할 수 있으나 결정적으로 이벤트의 종료 시점을 알 수 없어 속도 조절이 쉽지 않다.

이와 같은 기능 비교 결과로 판단해 보건데 웹킷의 기능 중 위 4가지에 있어서는 아이폰이 근소하지만 우수한 것으로 판단된다.

한편, 아이폰 하이브리드 웹 응용 프로그램은 아이폰 응용 개발의 진입장벽을 낮추고 기존의 웹 기술을 활용하여 간편하게 만들 수 있음을 보여주고 있다. 즉 개발

자가 아이폰 개발언어인 오브젝티브-C에 대한 지식이 없어도 HTML, CSS, 자바스크립트 등 웹 기술을 사용하여 GPS와 가속센서 등 아이폰에 내장된 기능을 활용한 응용을 만들 수 있는 것이다. 웹 응용프로그램으로도 화려한 UI나 애니메이션, 게임관련 웹 응용을 제작할 수 있으며, 쉽고 빠르게 개발 할 수 있어서 즉각적인 시장 트렌드를 반영할 수 있다. 또한 하이브리드 웹 응용 관련 자료도 많이 출판되고 있어 아이폰 응용프로그램 개발에 도움을 주고 있다[23]. [표 6]은 개발자 측면에서 안드로이드와 아이폰 개발 환경 비교 결과를 보이고 있다. 안드로이드는 다양한 기기에 탑재되어 이용하기 때문에 기본적인 성능만 지원하므로 아이폰과 비교 시 성능의 차이가 확연히 들어난다. 아이폰 개발은 단일 환경에서 서비스를 지원하기에 플랫폼 환경을 고려하지 않아도 되는 이점이 있다. 그러므로 독립적이고, 집중적인 엔진 개발 등 성능 지향에 초점을 맞추고 있지만 안드로이드는 오픈 플랫폼 개발 향상에 목적을 두고 있어 뛰어난 성능 지원 보다는 다양한 플랫폼 환경에 표준적인 지원을 염두하고 있다. 앞으로 아이폰 플랫폼도 다양한 이종 기기에 사용될 것을 고려하여 호환성에 관련된 지원도 필요할 것이다.

표 6. 개발자 측면 안드로이드와 아이폰 개발환경비교

안드 로이 드 개발 환경	개발 언어	자바에 친숙한 개발자에게는 개발이 용이함
	GUI 개발	기존에 사용되는 GUI 개발 관련 클래스와 비슷한 개념을 쓰고 있어 친숙 기존 경험을 활용하여 개발 가능
	API 활용	개발에 사용하는 API 완성도가 부족 (무분별한 상속 구조, 이벤트 전달 부실)
	UI 개발	레이아웃을 XML로 작성할 수 있으나 자바로 작성하는 것이 편리. 에디터에서 UI 스타일 반영 안됨
	에 물 레 이 터	실행 속도가 매우 느림
아이 폰 개발 환경	개발 언어	Objective C 언어에 익숙한 개발자에게는 용이함 내장형 응용 개발이 아닐 경우 웹 언어(HTML/CSS 등) 이용하여 웹 응용 개발 가능
	GUI 개발	기존 GUI 개발 관련 클래스와는 다른 UI 클래스 구조를 가지고 있음
	API 활용	API 완성도가 높고 활용 가능한 것, 활용 못하는 것에 대한 구분이 명확히 되어있음
	UI 개발	에디터에서 UI 스타일 확인 가능. 눈으로 직접 보면서 구성할 수 있음(인터페이스 빌더).
에 물 레 이 터	실행속도 빠름	

IV. 안드로이드와 아이폰의 웹킷 성능 비교

앞서 각 스마트폰에 탑재된 웹킷의 기능들을 비교한 결과 웹킷 탑재라는 공통점이 있지만 기능구현 측면에서 차이를 보임을 알 수 있었다. 특히 아이폰은 웹상에서도 쉽게 사용자 편의를 위한 인터페이스 구현할 수 있는 환경을 제공하기 때문에 개발자들이 편리하게 사용할 수 있는 강점이 있다. 이는 웹킷을 구성하는 자바스크립트 엔진에 따른 성능 차이 때문이다. 이번 장은 기본적인 웹킷 사양 및 성능에 대해 비교한다.

4.1 웹킷 사양

기본적으로 웹킷 사양은 [표 7]과 같이 요약 할 수 있다. 안드로이드와 아이폰 웹 브라우저는 웹킷을 기반으로 개발되어 대부분의 성능은 비슷하나 각각 자신들의 단말기 환경에 맞게 수정되었기 때문에 차이점이 있다.

표 7. 웹킷 기본 사양

	지원 정보
XML	XML 1.0 : 부분적 지원
Graphic	JPEG/JFIF : 85 버전 이상 GIF 89 a : 85 버전 이상
CSS	CSS 1: 85 버전 이상 CSS 2.1: 대부분 지원 CSS 3: 조금 지원
DOM	DOM level 1: 85 버전 이상 DOM level 2: 대부분 지원 DOM level 3: 조금 지원
JavaScript	JavaScript 1.5 extensions 지원

ACid3[23]은 웹 브라우저가 웹 표준을 얼마나 준수하고 있는지 검사하는 웹 표준 프로젝트의 테스트 페이지로 이를 이용하면 각 브라우저의 차이점을 명확히 알 수 있다. 특히 ACid3은 하이브리드용 웹 구현 시 스마트폰의 풍부한 UI구현을 도와주는 DOM과 자바스크립트에 관련된 준수 사항을 중점적으로 검사한다. 테스트를 통과하기 위해서는 100점을 획득해야 한다. [그림 3]은 안드로이드와 아이폰 브라우저의 ACid3 테스트 결과를 나타낸다. 아이폰에는 기존 웹킷에 있는 자바스크립트 엔진대신 스쿼릴피쉬(SquirrelFish)[24]라는 새로운 엔진을 장착했다. 기존 JavaScriptCore를 개선하여

실행성능을 향상 시킬 목적으로 개발되었으며, 이에 경쟁하듯 구글은 V8[25]이라는 신형 엔진을 개발하여 장착했다. 하지만 [그림 3]에 나타난 점수를 보면 안드로이드의 브라우저 성능이 아이폰보다 떨어진다는 것을 알 수 있다.



그림 3. Acid 3 테스트 모바일 브라우저 비교

4.2 웹킷 성능 비교 결과

앞서 보았던 Acid3 웹 브라우저 사양 비교 도구 외에도 최근 관련 회사들에서는 웹킷 엔진의 성능까지 측정할 수 있는 벤치마크(benchmark)를 함께 공개했다. [표 8]은 대표적인 벤치마크와 특징을 간단히 요약하고 있다.

[표 9]에서 볼 수 있듯이 자회사의 엔진 성능이 더 좋은 점수를 보였다. 주목할 점은 안드로이드나 애플에서 성능향상을 위해 많은 노력을 하고 있음을 알 수 있다.

표 8. 웹킷 성능 벤치마크 도구들

개발 회사	벤치 마크	특징
모질라	Dromaeo[26]	SunSpider와 자바스크립트엔진 기반으로 개발
구글	V8 Benchmark[27]	기존개발된 자바스크립트로 구현하여 사용
애플	SunSpider[28]	실제 사용되는 자바스크립트 코드 사용

표 9. 벤치마크별 자바스크립트 성능비교

Device	V8 Benchmark (Higher Score Is Better)	Sunspider, In Seconds (Lower Score Is Better)	Dromaeo (Higher Score Is Better)
Android 2.2	287.0	5.8	330.39
Android 2.1	65.0	14.3	270.07
iPhone 4	83.4	10.9	-
iPad	101.0	10.1	-
iPhone 3GS*	49.3	14.2	319.6
iPhone 3G*	23.0	41.4	-

지금까지 살펴보았듯이 대부분의 웹 응용이 아이폰 응용으로 제작되어 상용화되고 있는 실정인데, 이는 안드로이드용 웹 응용 개발 시 불편한 점이 많음을 보여주는 단적인 증거라 할 수 있다. 안드로이드폰에서 사용할 수 있는 웹 응용을 제작할 때는 개발자가 개발 프로그램 소스에 웹 브라우저에서 사용되는 모든 이벤트들에 대한 코딩을 일일이 해야 한다. 즉 안드로이드용 내장형이나 웹 응용을 개발할 때 사용자 편의를 위한 고려 사항을 개발자가 직접 코딩해야 하므로 제작 기간이 길어지며, 하루가 다르게 바뀌는 모바일 시장에서 사용자의 요구를 충족하는 콘텐츠를 제공하기 힘들다. 또한 안드로이드 하이브리드 웹 응용 제작 관련 자료나 서적이 아이폰에 비해 부족하다.

V. 결론

아이폰은 멀티 터치스크린의 강력한 기능, 세련된 디자인 그리고 다양한 소프트웨어를 갖추고 있다. 그러나 아이폰 응용은 애플 컴퓨터에서만 개발할 수 있으며 독점적이다. 애플이 자랑하는 아이튠즈와 앱스토어도 아이폰만을 위한 서비스이다. 이에 반해 구글은 독자적인 구글폰을 만들고 동시에 안드로이드 OS를 공개하였는데, 오픈 소스 기반의 안드로이드 OS는 구글이 제시하는 라이선스 조건만 따른다면 어떤 단말기 제조업체나 이동통신사도 안드로이드 OS 탑재 스마트폰을 만들어 팔 수 있다. 게다가 애플 앱스토어에 견주는 안드로이드 마켓도 진행하고 있다.

본 논문에서는 아이폰과 안드로이드폰의 웹킷 기능과 성능을 비교분석 하였다. 분석결과 아이폰은 응용을 개발하는 개발자와 사용하는 사용자 둘 다에게 편리함을 제공하고 있다. 이는 아이폰용 응용이 훨씬 다양하다는 점에서 안드로이드폰 웹 개발 시 개발자의 부담이 더 크기 때문인 것으로 판단된다. 따라서 결론적으로 웹킷 API 편리성, 우수성 측면에서 애플의 아이폰이 안드로이드에 비해 우수하다고 할 수 있다.

본 연구의 질을 더 높이려면 웹킷 기능 비교를 위한 표준화된 척도나 도구를 개발해야 한다. 그리하여 웹킷

엔진의 세세한 부분까지 모두 비교할 수 있게 해야 진정한 웹킷 비교 결과를 얻을 수 있을 것이다.

참고 문헌

- [1] 김민식, 정현준, “휴대폰 산업의 탈추격형 대응전략”, 정보통신정책, 제22권, 제1호, p.7, 2010.
- [2] 정제호, “플랫폼의 대 확장: 웹에서 모바일로”, SW Insight 정책리포트, pp.8-9, 2008(2).
- [3] 권기덕, “스마트폰이 IT시장에 미치는 영향”, SWInsight 정책리포트, 2009.
- [4] <http://www.apple.com/iphone>
- [5] www.openhandsetalliance.com
- [6] <http://www.freebsd.org>
- [7] <http://blog.naver.com/pskkmj/140014706849>
- [8] 에리카 세든, “예제로 시작하는 아이폰 개발”, 에어콘, 2009(2).
- [9] <http://www.cocoatouchappps.com>
- [10] <http://en.wikipedia.org/wiki/ANSLC>
- [11] <http://developer.apple.com>
- [12] <http://developer.apple.com/technologies/safari>
- [11] <http://www.htc.com>
- [13] <http://cellphones.org/blog/google-chrome-goes-mobile/>
- [14] www.konqueror.org
- [15] www.ajax.org
- [16] <http://dev.w3.org/html5/spec/Overview.html>
- [17] <http://en.wikipedia.org/wiki/Viewport>
- [18] “아이폰/아이팟 터치용 웹 애플리케이션 개발 팁 12개”, 2009. 12. 14.
- [19] <http://developer.android.com>
- [20] 에리카 세든, “예제로 시작하는 아이폰 개발”, 에어콘, 2009. 2.
- [21] 전종홍, 이승윤, “차세대 모바일 웹 애플리케이션 표준화 동향”, 전자통신동향 분석, 제25권, 2010(2).
- [22] developer.android.com/.../GestureDetector.html

[23] <http://webkit.org/blog/189/announcing-squirrelfish>
 [24] <http://code.google.com/intl/ko/apis/v8>
 [25] <http://www.acid3.acidtests.org>
 [26] <http://dromaeo.com>
 [27] <http://v8.googlecode.com/svn/data/benchmarks/v5/run.htm>
 [28] <http://www2.webkit.org/perf/sunspider-0.9/sunspider.html>

- 2008년 : 뉴욕주립대 스토니브룩 Research Scholar
 <관심분야> : Mobile System Software, Storage Systems, Computational Finance, Cluster Computing, Parallel and Distributed Operating Systems, Embedded System Software

저 자 소 개

이 고 은(Goeun Lee)

정회원



- 2007년 2월 : 숙명여자대학교 멀티미디어과학전공(이학사)
- 2010년 8월 : 숙명여자대학교 멀티미디어과학전공(이학석사)

<관심분야> : 모바일 소프트웨어, 스마트폰 이러닝

이 종 우(Jongwoo Lee)

정회원



- 1990년 2월 : 서울대학교 컴퓨터 공학과(학사)
- 1992년 2월 : 서울대학교 컴퓨터 공학과대학원(석사)
- 1996년 8월 : 서울대학교 컴퓨터 공학과대학원(박사)

- 1996년 ~ 1998년 : 현대전자(주) 정보시스템사업본부 과장
- 1998년 ~ 1999년 : 현대정보기술(주) 책임연구원
- 1999년 ~ 2002년 : 한림대학교 정보통신공학부 조교수
- 2002년 ~ 2003년 : 광운대학교 컴퓨터공학부 조교수
- 2003년 ~ 2004년 : 아이닉스소프트(주) 개발이사
- 2004년 ~ 현재 : 숙명여자대학교 이과대학 멀티미디어학과 부교수