

모바일 하이브리드앱 개발도구에 관한 비교연구

장래영* · 정성재** · 배유미* · 성경*** · 소우영*

*한남대학교 컴퓨터공학과, **마이호스팅, ***목원대학교 컴퓨터교육과

A Comparative Study For Mobile HybridApp Development Tool

Rae-Young Jang* · Sung-Jae Jung** · Yu-Mi Bae* · Kyung Sung*** · Woo-Young Soh*

*Hannam University, **Myhosting, ***Mokwon University

E-mail : rene402@hnu.kr, posein@paran.com, yumidw@hanmail.net, skyys04@mokwon.ac.kr,
wsoh@hnu.kr

요 약

애플(Apple) 아이폰(iPhone)으로 시작된 모바일 기기 보급은 구글(Google) 안드로이드(Android)가 등장하면서 확산되었다. 최근들어 모바일기기 범위는 아이폰과 안드로이드, 바다(Bada), 윈도우폰 7(WindowPhone7), 블랙베리(BlackBerry)등 스마트폰과 아이패드(iPad)와 안드로이드기반 태블릿(Tablet)등으로 다양해졌다. 과거에는 특정 기기에 맞는 전용 어플리케이션(Application, 이하 앱)만 개발하면 되었지만, 현재는 다양한 운영체제의 특성을 고려하여 개발하여야만 한다. 그에 따라 많은 문제점이 발생하고 기존 개발방법과 다른 개발방법들이 등장하였다. 본고에서는 이런 개발방법중 원소스멀티유즈(One Source Multi Use)를 지향하는 모바일 하이브리드앱(Mobile Hybrid App) 개발도구에 대해 알아보고 하이브리드앱 개발도구의 차이점을 비교하여 상황에 맞는 개발방법을 제시한다.

Abstract

Mobile devices diffusion begins by the iPhone and spreads by the Android. A widely used mobile devices is Smartphone(iPhone, Android, Bada, WindowPhone7, BlackBerry, etc) and Tablet(iPad, Android). In the past, we have developed a dedicated application for device. But, Now we must develop a application by considering the characteristics of the operating system. Accordingly, many problems occurred and other developmet methods have emerged. In this paper, we learn about MobileHybridApp development tool-oriented One Source Multi Use-and compare the differences, propose a method to develop context-sensitive.

키워드

NativeApp, HybridApp, WebApp, Titanium, PhoneGap, Appspresso

I. 서 론

최근 모바일기기 범위는 기존의 아이폰, 안드로이드폰을 비롯해 바다(Bada), 윈도우폰7(WindowPhone7), 블랙베리(BlackBerry), 심비안(Symbian)등의 스마트폰 계열과 아이패드(iPad) 및 안드로이드기반 태블릿(Tablet)계열로 확대되었다. 늘어난 모바일플랫폼과 더불어 안드로이드계열의 파편화는 애플리케이션개발에 문제점을 가져왔다. 출시 이후 빠른속도로 성장하여 아이폰 보급률을 뛰어넘어 모바일 OS시장의 1위 자리를 지키고 있는 안드로이드OS의 가장 큰 문제는 '파편화'다. 파편화란, 4.0(IceCreamSandwich)으로 통합이 되었지만 이전까지 2.3진저브레드(GingerBread)인 스마트폰OS와 3.0허니콤(HoneyComb)인 태블릿OS로 나뉘어

져 생긴 부작용을 뜻한다. 또한 구글의 권장사항이었지만 자유롭게 수정이 가능해 각 제조사별로 맞춤제작된 하드웨어와 OS는 (모토로라의 안드로이드폰은 권장해상도가 아닌 전용해상도) 이런 파편화 문제를 더 심각하게 만들었다. 이같은 기기간 다양한 OS버전, OS분리정책을 비롯해 맞춤화된 제조사별OS는 다양한 부작용을 일으켰다. 일부 기기에서 정상작동하는 애플리케이션이 다른 안드로이드기기에서 제대로 작동하지 않는 등 같은 안드로이드기기간의 호환성 문제가 지속적으로 발생하고 있다. 이런 문제점을 해결하려면 기존의 애플리케이션 개발과정을 살펴봐야한다. 예를 들어 기업소개애플리케이션을 제작하려하면 Objective-C와 JAVA를 이용하여 각각 아이폰과 안드로이드용으로 애플리케이션을 제작하여 앱스

토어(AppStore)와 구글플레이(GooglePlay)에 등록하여야 한다. Bada, WP7 및 다른 플랫폼까지 지원하려면 더 많은 시간과 비용을 소모해야한다. 업데이트내용이라도 있다면 처음부터 같은 작업을 반복해야 한다. 앞으로 더 많은 모바일 플랫폼이 발매된다면 그때그때 추가 작업을 해줘야 할 것이다.

표 1. 모바일플랫폼별 개발상황 비교

플랫폼	개발언어	개발툴	개발환경
Android	java	Android SDK Eclipse	Win/Linux/Mac
iPhone	Objective-C	X Code	only Mac
WP7	C#	Silverlight VisualStudio	only Win
Bada	C/C++	Bada SDK Eclipse	Win

표1 은 각 모바일플랫폼별 개발언어와 개발방식의 차이점을 비교/분석한 것이다. 표1 과 같이 모바일기기 플랫폼에 따라 습득할 언어, 설치해야할 개발툴, 개발에 필요한 환경구축방법등이 모두 다르고 이는 개발자들에게 많은 어려움과 불편함을 초래한다. 하이브리드기술은 이런 문제점의 해결책으로 등장했다. 특히 오픈소스인 웹킷과 HTML5의 발전은 하이브리드기술 등장에 영향을 미쳤다. 웹킷이란 웹브라우저를 만드는 기반기술이다.[1] 원래는 맥OS X에 탑재된 사파리에 사용하기위해 만들어졌으며 후에 오픈소스로 공개되었다. 현재 표2와 같이 대부분 모바일브라우저에서는 웹킷을 지원하고 있다. 하이브리드앱은 웹킷기반의 웹브라우저기술과 HTML5, CSS, JavaScript등을 통해 만들어진다.

표 2. 모바일플랫폼별 웹킷지원현황

구분	Android	iPhone	Bada
웹브라우저	사파리	크롬	돌핀
웹킷지원	○	○	○

본고는 하이브리드앱의 배경이 되는 모바일애플리케이션의 종류에 대해 알아보고 하이브리드앱 개발플랫폼들의 종류와 차이점을 비교하고 하이브리드앱 필요성과 각각 플랫폼별 활용법을 제시하도록 한다.

II. 모바일앱

일반적으로 최신기기에서 구동되는 모바일앱은 네이티브앱(NativeApp), 웹앱(WebApp), 하이브리드앱(HybridApp)으로 나뉜다.

2-1. NativeApp

모바일기기에 직접 설치되어 구동하는 전용애플리케이션이다. 각 플랫폼에 맞는 개발언어와 SDK로 개발하여야 하며 각 기기의 전용 리소스(주소록, 가속센서, 카메라 등)를 사용할 수 있다. 게다가 기기의 하드웨어, 메모리 등과 밀접하게 상호작용할 수 있어 모바일에 최적화된 고성능 애플리케이션을 개발할 수 있다. 또한 앱스토어나 구글플레이 같은 전용마켓을 통해 등록하고 판매할 수도 있다. 대부분 일반사용자들이 말하는 '앱(App)'은 네이티브앱을 지칭한다.

2-2. WebApp

최신 모바일 기기는 대부분 풀프라우징(Full Browsing)을 지원한다. 모바일웹앱은 브라우저에 종속되어 모바일에 더 최적화되어 구동되는 네이티브애플리케이션화된 방식을 말한다. 웹앱은 Web과 Application의 합성어로 모바일웹앱은 웹기술만 사용하여 폴스크린모드, 애니메이션 효과, 터치 인터페이스, 비동기 통신, 로컬 저장소, 오프라인 지원, 향상된 스타일 등을 구현하여 모바일 환경에서 네이티브앱과 유사한 실행환경, 사용자 경험(UX)을 제공하는 애플리케이션이다. 웹앱은 웹킷을 지원하는 브라우저를 통해 모든 디바이스에서 구동가능하다는 높은 호환성을 장점으로 갖는다. 네이티브앱과 달리 오류나 업데이트가 있더라도 비교적 짧은 시간에 수정/반영할 수 있어 유지보수도 용이하다. 그러나 브라우저의 속도와 성능에 의존적이고 특성상 서버와 통신이 주기적으로 이루어지다보니 실행속도가 느리고, 기기의 리소스를 손쉽게 이용할 수 없다는 것이 단점으로 지적된다.

2-3. HybridApp

하이브리드앱은 네이티브앱과 웹앱의 장점을 결합한 방식이다. 일반적인 웹기술로는 모바일 기기의 리소스를 직접 제어해 이용할 수 없다. 또한 일반사용자가 웹앱에 접근하려면 브라우저를 실행해 URL을 입력해 이용해야 한다. 즉 애플 앱스토어나 구글 플레이같은 마켓에 애플리케이션을 등록하거나 다운로드해 이용할 수 없다. 하이브리드앱은 이런 단점을 보완해 앱개발은 웹앱과 같이 웹기술을 이용해 손쉽게 개발하고 유지보수할 수 있고 네이티브앱처럼 모바일기기의 고유정보를 이용하고 앱스토어에 등록/판매도 할 수 있게 하는 애플리케이션이다. 이런 것이 가능하게 하려면 모바일기기의 리소스를 웹기술로 호출해 사용할 수 있는 추상화된 API가 필요하며 웹앱을 네이티브앱형식으로 감싸줄 수 있는 타이타늄(Titanium), 폰갭(PhoneGap), 앱스프레소(Appspresso) 같은 프레임워크가 필요하다. 표3 은 모바일앱의 종류를 비교해 나타낸 것이다.

표 3. 모바일애플리케이션 종류

	NativeApp	WebApp	HybridApp
특징	· 특정한 운영체제에 종속적 · 디바이스에 직접설치		· Native+Web
장점	· 최적의 UX 구현가능 · 빠른반응속도 · 리소스 이용가능 · 배포 용이 · 수익모델 창출 용이	· 브라우저 지원이 완벽하다면 모든 디바이스에서 이용가능한 높은 호환성 · 유지보수 용이	· One Source Multi-Use · NativeApp 보다 높은 개발생산성 · 배포 용이
단점	· 많은 개발 비용 · 각 플랫폼 별 개발비용 증가발생 · 유지보수 어려움	· 브라우저의 속도와 성능에 의존적 · 속도가 느림 · 리소스 이용 어려움	· 선택한 프레임워크에서 제공하는 것 이외의 기능 구현 어려움

III. 관련기술

웹으로 개발된 애플리케이션은 웹기술로 구현해 브라우저를 통해 구동하기 때문에 하나의 소스로 여러 플랫폼을 지원할 수 있다. 웹만의 장점인 손쉬운 업데이트 때문에 개발주기를 줄이고 사용자의 요구에 즉각적으로 반응할 수 있다. HTML5를 접목하면 오프라인 지원, 로컬저장소활용, 2D그래픽처리, 미디어재생, 소켓통신, 멀티쓰레드 프로그래밍 등의 구현도 가능하고 타이타늄, 폰갭같은 API를 활용해 네이티브앱같은 장점을 누릴수도 있다. 원소스멀티플랫폼을 실현할 수 있는 모바일웹앱과 하이브리드앱은 앞으로 모바일 앱개발의 추세가 될 것이다.

3-1. Titanium

Accelerator에서 개발한 하이브리드플랫폼인 타이타늄은 비교적 개발하기 쉬운 자바스크립트, HTML, CSS, PHP, Python, Ruby 등을 이용해 모바일, 태블릿, 데스크톱 애플리케이션을 개발할 수 있는 오픈소스 개발 플랫폼이다.[2] 타이타늄은 자바스크립트엔진을 내장하고 있다. 타이타늄은 자바스크립트, PHP와 같이 비교적 사용하기 쉬운 언어로 새로운 언어(Objective-C, JAVA 등)에 대한 학습 부담을 줄여주고 신속하게 개발할 수 있게 지원하는 것이 목적이며 가장 큰 장점이다. HTML, CSS를 이용하여 화면과 UI를 구성하므로 디자이너와 협업이 쉽고 개발 생산성이 높다. 타이타늄은 크게 모바일과 데스크탑부분으로 나누어 개발할 수 있으며 개발에 필요한 SDK도 모

바일용과 데스크탑용으로 구분되어 제공되고 있다. 타이타늄은 다른 하이브리드앱개발방식과는 다르게 웹기술을 이용하여 네이티브앱을 개발하는 방식을 지원한다. 이는 개발언어는 웹표준언어지만 결과적으로는 네이티브앱을 만들어준다. 타이타늄API를 이용하여 네이티브앱과 동일한 성능을 제공하는 장점을 갖는다. 개발도구로 Titanium Developer와 Titanium Studio 같은 개발도구를 제공하고 있다. 타이타늄은 네이티브앱을 직접 제어하기 때문에 각 플랫폼에 최적화된 화면을 만들 수 있고 속도도 웹뷰를 이용하는 방식에 비해 빠르다. 반면 플랫폼별로 사용할 수 있는 컨트롤러가 다르기 때문에 플랫폼별로 화면을 구성해야 하고 플랫폼별로 다른API를 사용해야 할 경우에는 플랫폼에 맞춰 따로 프로그래밍을 해줘야 하는 단점이 있다.

3-2. PhoneGap

폰갭은 2008년 iPhoneDevCamp에서 HTML로 만들어진 웹페이지를 아이폰앱으로 구동할 수 있음을 보여주며 주목을 끌었고 단지 웹페이지를 보여주는 것만이 아닌 아이폰 자체의 기능(카메라, 주소록, GPS등)을 사용하는 기술은 개발자들에게 큰 반향을 일으켰다. iOS 에 그치지 않고 Android, WP7, Bada, Symbian, WebOS, Blackberry 등 가장 많은 모바일플랫폼을 지원한다. 폰갭은 가속도센서, 카메라, 나침반, 주소록, 파일, GPS정보, 미디어, 알람, 스토리지에 대한 기능을 API로 제공한다.[3] 폰갭을 이용한 하이브리드앱 개발에서 화면부분은 주로 jQueryMobile, SenchaTouch 등의 UI프레임워크를 사용해야 한다. 폰갭은 배포되는 프레임워크패키지 용량이 매우 작고 따로 통합개발환경이나 도구 등은 제공하지 않는다. 개발사인 Nitosi가 Adobe에 인수된 후 DreamWeaver5.5에 폰갭프로젝트 기능을 추가한점은 고무적인 변화지만 폰갭 전용IDE는 아니기 때문에 여전히 불편함이 존재한다. Xcode, Eclipse같은 통합개발환경에 폰갭 라이브러리를 연결하여 개발을 해야 하기 때문에 기존SDK개발 환경에 어느 정도 지식이 있어야 된다는 것이 단점이다. 폰갭은 2011년 7월 정식1.0버전이 공개된 이후 현재는 1.7.0버전이 공개될 정도로 빠른 기능추가가 이루어지고 있다.

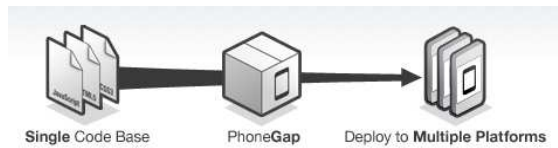


그림 1. 폰갭 개념도

3-3. Appspresso

KTH에서 개발한 플랫폼인 앱스프레스는 동작 방식이나 쓰임이 폰갭과 거의 동일하다. 웹뷰를

사용하고 디바이스에 관련된 API를 제공한다. 웹 앱으로 작성된 소스를 네이티브앱처럼 만들어 배포할 수도 있다. 폰갭은 독자적인 방식으로 제작된 API를 제공하는 반면 앱스프레소는 WAC표준 API를 제공하고 있다. WAC표준API를 이용하기 때문에 앱스프레소로 제작된 앱은 이동통신사가 제공하는 앱스토어에서 유통시킬 수 있는 장점이 있다. 앱스프레소 역시 앱스프레소IDE 통합개발환경을 제공한다. 타이타늄이 최근 Aptana를 인수하면서 통합개발환경을 제공하기전까지 하이브리드개발플랫폼중에서 최초로 통합개발환경을 구축했었다. 또 앱스프레소의 가장 큰 장점이라면 온더플라이빌딩('On the fly building')이다. 하이브리드 앱을 개발하면서 테스트를 하려면 매번 컴파일 과정을 거치게 되는데 온더플라이빌딩기능은 웹 앱부분만 수정이 되었을 때 실시간으로 실행중인 앱에 반영해주기 때문에 따로 컴파일과정을 거칠 필요가 없어 개발시간이 줄어들어 개발생산성을 높여준다. 또한 웹뷰방식이기 때문에 다양한 플랫폼을 지원할수 있다. 현재는 iOS, Android만 지원하고 있지만 KTH에서는 추후 WP7, Bada등으로 플랫폼을 지원해나갈 예정이다.[4]

IV. 하이브리드개발플랫폼 비교

크게 하이브리드개발플랫폼은 스크립트엔진을 사용하는 Titanium과 웹뷰(Webview)를 이용하는 PhoneGap, Appspresso 로 나눌수 있다. 폰갭, 앱스프레소등은 모바일기기 리소스에 접근할 수 있는 API가 제한되어 있고 개발자가 추가로 개발하여 사용할 수 없다. 네이티브앱에서 사용하던 기능이 전용API에서 제공하지 않는다면 개발자는 다른 방식을 고려하거나 개발을 포기할 수 밖에 없다. 그에 비해 타이타늄은 필요한 기능을 JAVA, Objective-C등을 이용해 개발하여 사용할 수 있는 확장성을 제공한다. 타이타늄이 제공하지 않는 모듈이라도 직접 모듈을 제작하여 사용하거나 다른 개발자들이 제작하여 공개한 모듈을 구매 사용할 수 있도록 지원하고 있다. 타이타늄의 오픈마켓플레이스에는 현재 250여개의 모듈이 등록되어 있어 원하는 기능을 쉽게 추가하여 사용할 수 있다.[5] 타이타늄은 Mobile SDK를 이용하면 HTML5를 지원하는 웹 앱을 개발할 수 있다. 폰갭같은 플랫폼은 jQueryMobile, SenchaTouch 등의 UI프레임워크를 이용해 별도로 개발하여야 한다. 모바일플랫폼의 지원에서도 차이가 있다. 폰갭은 대다수의 모바일플랫폼을 지원하는 반면 타이타늄과 앱스프레소는 iOS와 Android 만 지원하고 있다. 다음은 각 개발플랫폼별 차이점을 간략히 표로 나타낸 것이다.

표 4. 하이브리드개발플랫폼별 비교

구분	Titanium	PhoneGap	Appspresso
지원 플랫폼	iOS, Android	iOS, Android, WP7, Bada, BlackBerry	iOS, Android
개발 환경	Titanium Studio	각각 SDK에 라이브러리로 연결	Appspresso IDE
동작 방식	Script (Native Code Convert)	WebView	WebView
대상	Mobile/Desktop	Mobile	Mobile
제공 API	고성능 독자API 가능추가 가능	독자적인API 기능추가 불가	WAC표준API

V. 결론

지금까지 모바일 하이브리드 앱 개발도구에 관해 알아보았다. 각 개발도구는 각각의 특징이 명확해 어떤 것이 월등하게 낫다고 말하기는 어렵다. 상황에 맞는 개발도구를 선택하는 것이 가장 적절할 것이다. 가령 다양한API를 이용해 모바일 기기 리소스를 컨트롤하고 빠른실행을 원한다면 타이타늄, WAC표준API를 이용하고 다양한 앱스토어를 이용하고 싶다면 앱스프레소, 다양한 플랫폼을 지원하고자 한다면 폰갭, HTML5를 이용해 웹 앱형태로 개발하고 이를 네이티브화하려면 폰갭이나 앱스프레소를 사용하면 될 것이다. 향후 멀지 않은 미래에는 스마트폰뿐만 아니라 각종 모바일기기를 위한 통합개발플랫폼이 제시될 것이다. 개발자입장에서 플랫폼에 신경쓰지 않고 개발할 수 있는 환경이 구현될것이지만 아직까지는 그런 요구에는 하이브리드앱이 최선으로 꼽히고 있다. 상황에 맞춰 개발계획을 세워 진행한다면 보다 적은 투자비용으로 최대효과를 기대할 수 있을 것이라 여겨진다.

참고문헌

- [1] WebKit, <http://www.webkit.org>
- [2] Titanium, <http://www.appcelerator.com>
- [3] PhoneGap, <http://www.phonegap.com>
- [4] Appspresso, <http://www.appspresso.com>
- [5] Titanium Marketplace, <https://marketplace.appcelerator.com/>
- [6] 박기환 외 공저, "하이브리드 모바일 앱 개발을 위한 PhoneGap", 위너북스, 2012년
- [7] 방현우, "티타늄을 활용한 스마트폰 앱 개발", 위키북스, 2012년
- [8] 박중명, "모바일 웹 개발", 위키북스, 2012년