

논문 2013-08-02

이기종 스마트 플랫폼 상에서의 하이브리드앱 기반 스마트러닝 콘텐츠 호환성에 관한 연구

(A Research of the Compatibility for the HybridApp-Based
Smart-Learning Contents in the Heterogeneous Smart
Platform)

국종진*, 박병하

(Joongjin Kook, Byoung-Ha Park)

Abstract : With the development and general use of a variety of Android/iOS-based smart phones and smart pads, the existing e-learning contents need to be changed in such a way that they can be carried out on different smart device platforms. This paper shows what changes are needed for that aim, and, in particular, for the compatibility of different platforms by designing and implementing Android/iOS-based smart learning contents in the form of a hybrid app. This paper will hopefully help you consider what elements are required to develop smart-learning contents on a variety of platforms for mobile devices.

Keywords : Smart-learning, E-learning, Hybrid App

1. 서론

기존의 이러닝 콘텐츠는 대부분 웹 기반으로 작성되었다. 이는 콘텐츠의 호환성을 위한 것으로 인터넷의 사용이 가능하고, 인터넷 익스플로러가 설치되어 있으면 누구나 손쉽게 콘텐츠를 사용할 수 있도록 하였다. 또한 학습자와의 상호작용을 위해 자바스크립트 등의 스크립트 언어를 사용하며, SCORM이나 CC 기반의 학습관리시스템(LMS), 콘텐츠관리시스템(LCMS) 등과 연계되어 학습자의 학습 정보를 관리한다 [1]. 하지만, iOS 기반의 아이폰, 아이패드, 그리고 2008년 출시된 안드로이드 기반의 다양한 스마트폰과 스마트패드가 대중화되기 시작하면서 기존의 데스크톱 환경과는 달리 Webkit 기반의 브라우저를 사용하고 기기의 해상도가 저마다 다른 스마트 기기에서 기존의 콘텐츠를

그대로 사용하는 것은 거의 불가능하게 되었다.

특히 최근에 HTML5의 출시와 이를 기반으로 한 콘텐츠로 인해 이러닝 콘텐츠를 스마트 기기까지 확산시킬 수 있게 되었지만, 동일한 Webkit 기반의 웹 브라우저를 가진 플랫폼이라 하더라도 동일한 콘텐츠에 대해 완전한 호환성을 보장할 수 없으며, 해당 플랫폼의 네이티브 환경을 사용하기 위해서는 PhoneGap [2]이나 Titanium Mobile [3]과 같은 크로스플랫폼을 지원하는 프레임워크의 사용이 불가피하다. 또한 크로스플랫폼을 지원하는 프레임워크를 사용한다 하더라도 상이한 스마트 플랫폼에서 동일한 콘텐츠에 대해 완벽한 호환성을 보장하지는 못한다.

본 논문에서는 가장 대중적인 iOS와 안드로이드에 대해 하이브리드앱 형태의 콘텐츠 뷰어를 설계하고 구현함으로써 상이한 플랫폼 환경에서 호환성 이슈가 발생하는 부분을 확인하고 비교해 보았다. 또한 호환성 문제의 원인파 해결 방안을 제시함으로써 향후 이기종 스마트 기기에 대한 스마트러닝 콘텐츠의 개발을 위해 필요한 소프트웨어 설계 방법에 대한 가이드라인을 제시하고자 한다.

* Corresponding Author (tipsiness@gmail.com)

Received: 30 Oct. 2012, Revised: 01 Dec. 2012,

Accepted: 18 Jan. 2013.

J. Kook, B.H. Park: KETI

※ 본 연구는 지식경제부 산업융합원천기술개발사업의 연구비 지원으로 수행되었음.

II. 스마트러닝 하이브리드앱의 구조

1. 하이브리드앱(HybridApp)

스마트 플랫폼에서 제공하는 SDK로 제작한 애플리케이션을 네이티브(native)앱이라고 한다. iOS 기반의 애플리케이션은 Objective-C로, 안드로이드 기반의 애플리케이션은 Java로 개발해야 한다. 이렇게 플랫폼별로 개발 언어와 환경이 다르기 때문에 다양한 화면 크기와 해상도를 갖는 이기종 플랫폼을 지원하려면 디바이스와 플랫폼의 개수만큼 시간과 노력을 들여야 하는 문제가 발생한다.

물론 사용자에게 제공하려는 기능을 표준 웹으로 구현할 수 있다면 애플리케이션 개발 노력을 줄일 수 있다. 하지만 웹브라우저에 구현할 수 있는 UX와 기능은 아직 네이티브앱에 비하여 열악하다. 결정적으로 표준 HTML과 자바스크립트로는 카메라나 가속도센서를 비롯한 스마트기기의 다양한 기능을 사용할 수 없다.

기존의 이러닝 콘텐츠는 웹 기반의 HTML, 자바스크립트, CSS 등을 사용하여 제작되었으며, 따라서 인터넷 익스플로러와 같은 웹브라우저를 통해 표현이 가능했다. 하지만 이러한 데스크톱 기반의 이러닝 콘텐츠는 인터넷 익스플로러와는 다른 Webkit을 사용하는 브라우저에서는 호환성이 결여된다. 이러한 문제점을 보완하기 위해 HTML5를 기반으로 한 하이브리드앱의 개념이 등장하였다.

하이브리드앱을 개발할 때에는 터치스크린을 사용하는 모바일 기기에 특화된 프레임워크가 쓰인다. Sencha Touch [4]는 아이폰과 안드로이드에서 HTML5 기반 모바일웹으로 앱을 만들 수 있는 자바스크립트 프레임워크이다.

jqtouch [5]는 자바스크립트 프레임워크 jQuery와 함께 사용하는 터치 인터페이스 플러그인으로 사용과 수정이 간편한 것으로 알려져 있다. 모바일 브라우저 주소창과 메뉴를 가려 앱처럼 보이게 만드는 것이 주요 기능으로 별도 패키징틀이 필요하다.

PhoneGap은 웹 기반 앱을 여러 스마트폰 플랫폼 기반의 일반 모바일 앱으로 감싸주는 오픈소스 플러그인 기술이다. 통합 개발 환경(IDE)이 없으므로 iOS 기반의 앱을 만드는 경우 objective-c를 사용해야 한다. 전용 지원 플랫폼은 아이폰, 안드로이드, 블랙베리, 심비안, 팜OS 등이다. 플랫폼 별로 차이는 있지만 대부분의 환경에서 지오로케이션(Geo Location), 진동, 가속도계, 소리, 단말기 주

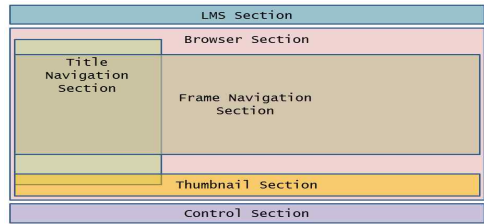


그림 1. 스마트러닝 하이브리드앱 구조

Fig. 1. Smart-Learning HybridApp Structure

표 1. 스마트러닝앱의 구성요소

Table 1. Smart-Learning App Elements

콘텐츠 영역	용도
LMS	학습 관리 시스템
Browser	HTML5 콘텐츠 표시
Frame Navigation	프레임 기반 콘텐츠 이동
Thumbnail	썸네일 기반 콘텐츠 이동
Title Navigation	제목 기반의 콘텐츠 이동
Control	콘텐츠와 앱 제어

소록 API를 다룰 수 있다.

Appcelerator의 Titanium은 웹개발자들이 자바스크립트, CSS, HTML 등의 웹 기술과 루비, 파이썬 같은 스크립트 언어를 사용해 아이폰, 안드로이드용 앱을 개발할 수 있게 해주는 플랫폼이다.

지금까지 나열한 크로스플랫폼용 프레임워크들은 공통적으로 HTML5 기반 웹프로그램의 일부 요소를 네이티브앱의 형태로 표현해 주기 위한 기능을 제공한다. 하지만, 스마트러닝을 위한 하이브리드앱은 콘텐츠 자체에서 발생하는 웹 기반 액션을 네이티브앱을 통해 처리해야 하기 때문에 기존의 프레임워크가 제공하는 기능을 통해 처리할 수 없다.

2. 스마트러닝 하이브리드앱의 구조

스마트러닝을 위한 콘텐츠의 호환성 테스트를 위해 그림 1과 같은 형태의 스마트러닝 하이브리드앱을 설계하였다.

그림 1의 스마트러닝앱은 LMS, Browser, Control, 그리고 몇 개의 Navigation 영역들로 구성되며, 각 영역의 용도는 표 1과 같다.

이러한 형태의 스마트러닝앱에서 호환성과 관련된 문제를 고려해야 하는 부분은 *Browser Section*과 *LMS Section*이다. 콘텐츠는 HTML5로 제작되

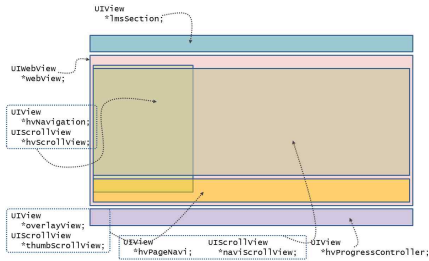


그림 2. iOS 기반 스마트러닝앱 구조
Fig. 2. iOS-based Smart-Learning App Structure

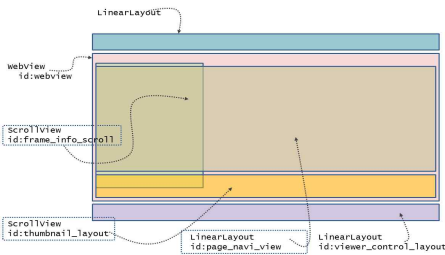


그림 3. 안드로이드 기반 스마트러닝앱 구조
Fig. 3. Android-based Smart-Learning App Structure

이 *Browser Section*에 표시되며, 인터랙티브한 학습을 위해 학습자의 액션이 필요한 부분은 자바스크립트를 사용하여 처리된다. 또한 진도 처리와 같이 LMS와 관련된 기능의 처리는 *LMS Section*에서 이루어진다. 두 영역에서 요청되는 사용자의 액션은 자바스크립트를 비롯한 웹 관련 기술을 통해 서버와의 통신이 필요하고, 이 과정에서 관련 데이터를 처리하기 위해 XML 형식의 데이터를 주고받을 수 있다.

3. iOS와 안드로이드 기반 하이브리드앱 설계

iOS 기반의 하이브리드앱을 구현하기 위한 소프트웨어의 설계는 그림 2, 안드로이드를 위한 소프트웨어 설계는 그림 3에 나타내었다. 그림 2, 3에서 화면의 구조는 완전히 동일하지만, *Browser Section*을 제외한 나머지 영역은 네이티브앱의 형태로 설계하였다.

그림 2, 3에서 나타내는 것처럼 iOS와 안드로이드 기반의 두 앱은 HTML5로 작성된 동일 콘텐츠를 사용할 것이며, 콘텐츠와 앱의 제어권을 위한 부분은 네이티브앱으로 처리한다. 또한 이 과정에서

표 2. 동작 유형
Table 2. Action Type

동작 유형	설명
LMS	학습자와 관련된 LMS/LCMS 처리
LNK	콘텐츠의 하이퍼링크 처리
PDF	부교재로 사용되는 pdf 파일의 내용 출력
VID	동영상 재생

PhoneGap이나 Titanium 등의 크로스 프레임워크를 사용하지 않았다.

III. 하이브리드앱 기반의 스마트러닝 콘텐츠 호환성

1. iOS와 안드로이드 기반 콘텐츠의 차이점

iOS와 안드로이드 기반으로 각각 구현된 스마트러닝 콘텐츠에서 호환성에 문제를 일으키는 부분은 콘텐츠로부터 데이터를 가져오는 부분과 미디어 플레이어 실행, 제스처에 대한 처리, XML에 대한 JSON 파서, 그리고 앱을 시작시킬 때의 처리 방식 등이다.

1.1 콘텐츠 내부의 학습자 행위의 처리

일반 이러닝 콘텐츠에서는 사용자의 행위에 대한 처리를 위해 자바스크립트와 같은 스크립트 언어를 사용한다. 이 때, 사용자의 액션에 대해 하이퍼링크를 처리한다거나, 레이어 형식의 별도의 뷰(view)를 팝업시켜 표시한다거나, 또는 LMS와 관련한 진도 처리를 하기 위해서는 콘텐츠 자체에 포함된 자바스크립트를 사용하여 처리하는 것은 불가능하기 때문에 별도의 프로토콜을 정의하여 네이티브앱을 기반으로 처리되어야 한다. 이러한 프로토콜을 정의하기 위해서는 기존의 웹기반 이러닝 콘텐츠에서 발생 가능한 동작의 유형들을 추출해야 하며, 요청된 동작에 따른 작업의 수행 방법과 필요한 인자들의 설정이 필요하다.

이러닝 콘텐츠에서 다루어져야 하는 대표적인 동작의 유형은 표 2와 같이 정의될 수 있다.

표 2와 같은 이러닝 콘텐츠의 주요 기능을 수행하기 위해서는 웹기반 콘텐츠의 요구사항을 네이티브앱으로 전달할 수 있어야 한다. 즉, 웹(Web)과 앱(App) 간의 인터페이스가 필요하며, 아직까지 하이

브리드앱에서 요구하는 웹과 앱 간의 표준화된 인터페이스는 없다.

본 논문에서는 하이브리드앱을 위해 필요한 웹과 앱 간의 인터페이스를 HWI(*HybridWeb Interface*)로 정의하였으며, 형식은 다음과 같다.

```
interface HybridWebInterface {
    type interface_name (arguments, ...);
    ...
}
```

HWI 인터페이스를 기반으로 표 2의 동작 유형을 정의하면 다음과 같이 표현될 수 있으며, 실제 구현을 위해서는 자바스크립트 등의 스크립트 언어를 사용할 수 있다.

```
interface HybridWebInterface {
    LMS lms(user_info, content_info, state);
    LNK link(url);
    PDF pdf(url or path);
    VID video(url or path);
}
```

1.2 제스처(gesture) 처리

iOS는 제스처의 처리를 위해 제스처 프레임워크를 제공한다. iOS의 제스처 프레임워크에서는 *touch*, *swipe*, *pinch*, *drag* 등의 일반적인 제스처 처리에 대한 API를 포함하고 있지만, 안드로이드에서는 iOS와 같은 유형의 제스처 프레임워크가 제공되지 않기 때문에 터치 이벤트의 시작과 끝에 대한 계산을 통해 제스처를 처리해야 한다.

1.3 XML 처리

LMS와 LCMS 기능을 처리할 때와 같이 서버와 데이터 교환을 하는 과정에서 XML 형식의 데이터를 주고받으며, 이 때 필요한 파라미터들은 JSON 형식을 따른다. 따라서 데이터를 서버로 전송하는 과정에서 JSON 형식의 XML 데이터가 만들어져야 하며, 서버로부터 수신한 데이터를 처리하는 과정에서 JSON 파싱(parsing)이 필요하다.

안드로이드에서는 JSON에 대한 처리를 위해 *org.json.JSONException*, *org.json.JSONObject* 클래스가 제공된다. 또한 http 요청을 처리하기 위해 *HttpParams*, *HttpProtocol*, *HttpClient*, *HttpGet*, *HttpResponse* 등의 클래스가 제공된다. 반면 iOS에서는 *ASIHTTPRequest*, *ASIFormDatRequest* 오브젝트를 사용하여 JSON에 대한 처리를 수행할 수 있다.

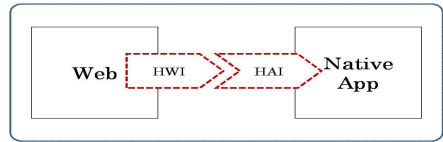


그림 4. 하이브리드앱의 인터페이스 구조
Fig. 4 HybridApp Interface Structure

1.4 앱 시작 방식

콘텐츠에 대한 접근이 웹을 통해 이루어지는 경우 iOS와 안드로이드 각각에서 발생하는 앱의 실행 요청 방식이 다르다. 안드로이드는 해당 앱을 가리키는 *Intent* 오브젝트가 사용되고, 파라미터에 대한 전달은 *Intent extra*를 통해 이루어지는 반면, iOS에서는 애플리케이션이 시작될 때 *AppDelegate*를 통해 호출되는 *didFinishLaunchingWithOptions* 메서드를 사용함으로써 애플리케이션이 시작될 때 전달되는 인자를 처리할 수 있다.

2. *HybridAppInterface* 기반 콘텐츠 호환성 평가

앞서 나열한 두 플랫폼간의 차이점을 고려하여 HTML5로 제작된 기존의 콘텐츠에 HWI를 추가하고, 각각의 동작 유형에 대해 iOS와 안드로이드에서 동일한 동작 결과를 보이는지 실험해 보았다.

HWI는 기존의 웹기반 콘텐츠에 삽입되는 인터페이스 정보이기 때문에 실제 기능을 수행하기 위해서는 플랫폼마다 독립적으로 네이티브앱 형태의 구현이 필요하다. HWI와 대응되는 네이티브앱의 인터페이스를 HAI(*HybridAppInterface*)로 정의하였으며, HWI와 HAI를 기반으로 한 하이브리드앱의 구조는 그림 4와 같다.

안드로이드와 iOS를 비롯한 많은 플랫폼에서 웹 기반 콘텐츠의 접근을 위해 *WebView* 오브젝트가 사용된다. 따라서 HAI의 구현을 위해서는 *WebView* 클래스의 상속을 바탕으로 크로스플랫폼을 위한 공통의 인터페이스를 만드는 것도 가능하다. 안드로이드에서는 각각의 동작 유형에 대한 처리를 위해 *WebView* 오브젝트의 *WebViewClient* 인터페이스를 사용하여 동작 유형을 확인함으로써 사용자의 요청에 대한 앱 기반의 처리를 수행할 수 있다. 반면 iOS에서는 *WebView* 오브젝트의 *shouldStartLoadWithRequest* 메서드를 사용하여 콘텐츠에서 발생하는 자바스크립트의 이벤트를 가로채기 함으로써 학습자의 요청을 처리하도록 하였다.

iOS와 안드로이드에서 스마트러닝앱의 시작을

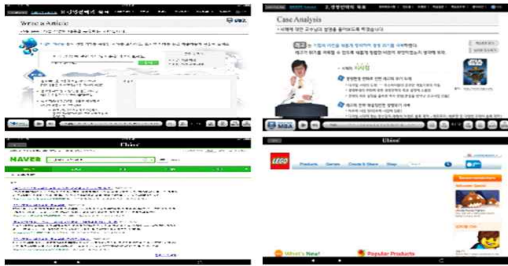


그림 5. 동작 유형 LNK(좌:안드로이드, 우:iOS)
Fig. 5 Action type:LNK(left:Android, right:iOS)



그림 6. 동작 유형 VID(좌:안드로이드, 우:iOS)
Fig. 6 Action type:VID(left:Android, right:iOS)

비롯하여 LMS, LNK, PDF, VID 동작에 대한 인터페이스를 구현하고 실행 결과를 확인했을 때, 두 플랫폼 모두에서 그림 5, 6과 같이 동일한 실행 결과를 확인할 수 있다.

그림 5의 실행 결과에서 만약 콘텐츠에서 하이퍼링크에 대한 처리를 위해 기존의 웹기반 콘텐츠와 같이 `<a href>` 태그를 사용한다면, 해당 링크에 대한 처리는 현재 화면에서 이루어지거나 내장된 브라우저를 호출하여 처리하게 된다. 하지만, 위의 결과와 같이 뷰가 중첩되는 구조를 통해 하이퍼링크를 처리하고자 한다면 기존의 HTML과 자바스크립트만으로는 처리가 불가능하다.

그림 6은 VID 동작에 대한 실행 화면을 나타내는데, iOS와 안드로이드가 웹 관련 기능을 다루기 위해 공통적으로 *WebView*를 사용하지만 동영상 처리에 대한 방식은 다소 차이를 보인다. iOS는 HTML5에 포함된 동영상을 *iframe*을 통해 현재 화면에서 바로 재생이 가능하지만, 안드로이드는 이를 지원하지 않기 때문에 *VideoView*에 대한 별도의 클래스를 만들고 인텐트 오브젝트를 사용해야 내장된 동영상 플레이어를 통해 동영상 재생이 가능하다.

이와 같이 플랫폼간의 차이점으로 인해 웹 기반의 콘텐츠에서 발생하는 사용자의 요청을 순수하게 웹의 기능만을 사용하여 처리하는 것은 불가능하며, 따라서 웹 기반의 콘텐츠에서 발생 가능한 여러 가지 유형의 요청에 대해 웹과 앱 간의 인터페이스를 정의하면 이기종 플랫폼에 대해 쉽게 동일한 콘텐츠를 제공할 수 있게 된다. 표 2의 동작 유형 이외의 다른 동작 유형에 대해서도 웹 기반의 HWI와 네이티브앱 기반의 HAI 인터페이스를 정의하고 네이티브앱에서는 해당 요청을 처리할 수 있는 *WebView* 기반의 템플릿을 구성하게 되면 기존의 이러닝 콘텐츠를 스마트러닝을 위한 콘텐츠로 쉽게 이식할 수 있을 것이다.

IV. 결 론

본 논문에서는 가장 대중적인 안드로이드와 iOS에서 동일하게 표현되는 스마트러닝앱을 설계하고 구현해 봄으로써 스마트러닝을 위한 하이브리드앱의 플랫폼 종속성을 분석하고, 웹 기반의 HWI와 네이티브앱 기반의 HAI 인터페이스를 설계하여 각 플랫폼 별로 해당 기능을 구현해 보았다. 이를 통해 기존의 웹 기반 이러닝 콘텐츠를 스마트기기에서 사용할 때의 플랫폼에 종속적인 제약 사항을 도출할 수 있으며, 웹과 앱 간의 간단한 인터페이스를 정의함으로써 크로스 플랫폼에 대응 가능한 스마트러닝 콘텐츠의 구현이 가능함을 확인할 수 있다.

참 고 문 헌

- [1] R. Ellis, R.A. Calvo, "Minimum indicators to quality assure blended learning supported by learning management systems," *Journal of Educational Technology and Society*, Vol. 10, No. 2, pp.60-70, 2007.
- [2] <http://phonegap.com/>
- [3] <http://www.appcelerator.com/>
- [4] <http://www.sencha.com/products/touch>
- [5] <http://www.jqtouch.com/>

저 자 소 개

이 기 종



2005년 광운대 컴퓨터공학과 학사.

2007년 광운대 컴퓨터공학과 석사.

2012년 숭실대 컴퓨터공학과 박사.

현재, 전자부품연구원 박사후연구원.

관심분야: 임베디드 소프트웨어, 센서네트워크, 스마트러닝, 크로스플랫폼.

Email: tipsiness@gmail.com

박 병 하



1999년 세종대 전자과학과 학사.

2001년 세종대 컴퓨터공학과 석사.

현재, 전자부품연구원 책임연구원.

관심분야: 감성공학, 센서네트워크, 스마트러닝.

Email: bhpark@keti.re.kr