

멀티 플랫폼 지원을 위한 웹 기반 통합개발환경 구현

*이명춘 신사임 장달원 이종설 장세진

전자부품연구원

*lmcspring@keti.re.kr

Implementation of Web based Integrated development environment for Supporting Multi Platform

*Lee, MyeongChun Sin, Saim Jang, Dalwon Lee, JongSeol Jang, Sei-Jin

Korea Electronics Technology Institute

요약

본 논문은 개발자에게 멀티플랫폼 지원을 위한 웹언어 기반의 하이브리드 개발환경을 제안한다. 최근 웹 기술의 발달로 웹 언어 기반의 애플리케이션이 증가 하고 있다. 그러나 웹 언어로 구현된 모바일 애플리케이션의 경우 디바이스의 플랫폼에서 제공하는 고유의 자원들에 대한 사용이 어려우며, 애플리케이션 마켓에서 배포가 어려운 단점이 있다. 본 논문에서는 이러한 문제를 해결하기 위해 통합개발환경을 구현하여 개발자들이 웹언어로 다양한 플랫폼에서 애플리케이션을 개발할 수 있는 환경을 제공한다. 제안하는 통합개발환경은 안드로이드와 타이젠 운영체제를 지원하며, 브라우저 기반의 웹 에뮬레이터와 애플리케이션 패키징 인터페이스를 지원하여 개발되는 소스코드를 테스트할 수 있도록 한다. 또한, 웹 에뮬레이터에서 디바이스 고유의 리소스를 사용하기위한 API를 설계한다. 제안하는 방법의 검증은 위해 타이젠 기반의 IVI 디바이스와 안드로이드 스마트폰을 사용하여 웹 언어로 작성된 애플리케이션을 테스트 한다.

1. 서론

과거 웹 기술은 브라우저를 지원 하는 PC에만 국한되었지만 최근 모바일과 통신 기술의 발달로 PC 뿐만 아니라 스마트폰, 스마트패드 등에 웹 기술이 확산되어 적용되고 있다. 이에 따라 스마트 디바이스를 지원하기 위한 웹 기술들 또한 다양해지고 있으며, 최근 하이브리드 앱의 출현으로 웹 개발자들의 모바일 애플리케이션 개발에 접근성이 높아지고 있다[1].

모바일 웹 앱은 HTML과 같은 웹 언어로 개발된 애플리케이션을 의미하는 것으로 브라우저를 기반으로 실행되기 때문에 브라우저를 지원하는 거의 모든 기기에서 사용이 가능하다. 최근 HTML5의 출현으로 그 활용성이 증가하고 있다. 네이티브 앱은 기존 앱스토어에서 설치된 형태의 앱을 의미하는 것으로 디바이스의 모든 자원을 사용할 수 있지만, 하나의 플랫폼에서만 사용이 가능한 한계가 있다. 이는 다양한 플랫폼을 지원하는 앱을 개발하는 기업 입장에서는 시간과 비용이 상승하는 원인이 된다[2]. 하이브리드앱은 웹앱과 네이티브 앱을 혼합한 형태로 웹앱처럼 디바이스의 플랫폼 환경에 의존하지 않고 네이티브 앱처럼 카메라, 통화, 가속도 센서 등 디바이스의 플랫폼 리소스를 사용할 수 있다. 이는 앱을 개발하기 위한 비용을 줄일 수 있을 뿐만 아니라 웹 개발자들의 스마트 디바이스 애플리케이션 개발에 진입장벽을 낮출 수 있다. 그리고 앱의 형태 및 적용되는 기술 또한 다양화 될 수 있다.

그러나 이를 지원하기 위한 개발통합환경(IDE)는 아직 부족한 실정이다. 따라서 본 논문에서는 하이브리드앱 개발을 지원하기 위한 통합개발환경을 구현한다. 개발하는 통합개발환경은 사용자 프로젝트 관

리, 에뮬레이터, 앱 패키징 등의 기능을 구현할 뿐만 아니라 플랫폼 자원과 웹 자원을 지원하기 위한 에뮬레이터 지원 API를 설계 하여 기술적 사항을 제안한다. 본 IDE에서 지원하는 운영체제는 안드로이드와 타이젠으로 이를 위해 Crosswalk을 사용한다. Crosswalk은 오픈소스 기반의 웹 런타임으로 WebRTC, WebGL, Vibration API, Presentation API, WebView를 지원하며, C/C++로 작성된 소스코드를 웹언어 API로 구현할 수 있는 확장 기능을 제공한다[3].

본 논문의 구성은 다음과 같다. 먼저 IDE 개발을 위한 이클립스 플러그인에 대해 소개한다. 다음으로 IDE 구현 방법 및 각 기능에 대해 소개하고 에뮬레이터 환경에서의 API 설계에 대한 설명을 한다. 그리고 실행 디바이스에 관한 설명과 테스트를 통해 제안하는 방법을 검증한다.

2. 이클립스 플러그인

그림 1은 이클립스의 구조를 나타낸 것으로 이클립스 플랫폼의 IDE는 플러그인의 조합으로 이루어져 있다[4].

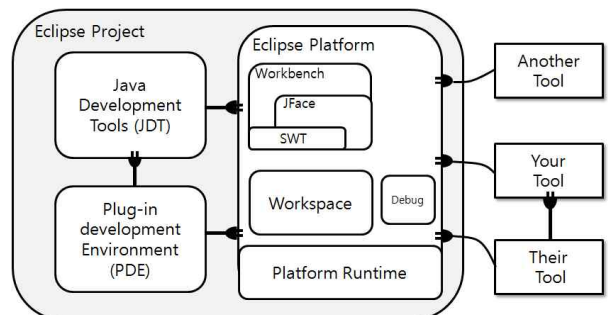


그림 1. 이클립스 구조

이클립스는 마이크로소프트의 비주얼스튜디오와 함께 개발자들 사이에서 가장 많이 사용되는 도구로 자바를 위주로 C/C++, 파이썬, 웹언어 등 다양한 언어를 지원하고 UML, Maven, CVS, Git 등의 개발 보조 도구를 제공한다. 초기와 달리 현재는 OSGI의 도입으로 범용 응용 소프트웨어 플랫폼으로 발전하였으며, 윈도우, 리눅스, Mac 등 대부분의 운영체제에서 사용이 가능하다. 또한, 이클립스 플러그인을 개발을 통해 다양한 환경을 지원하는 개발환경 도구를 제공할 수 있다.

3. 제안하는방법

이클립스 플러그인은 확장(Extension)과 확장점(Extension) 개념을 도입하여 플러그인간의 연동을 유연하게 한다. 본 논문에서 제안하는 IDE는 이클립스 플러그인 개발환경에서 제공하는 확장점을 이용하여 위자드, 뷰, preference 등을 목적에 맞게 구현한다. 하이브리드앱 개발 지원을 위한 기능은 프로젝트의 생성, 워크벤치 기능, UI에디터, 에뮬레이터, 앱 패키징으로 분류할 수 있다. 아래의 그림 2는 프로젝트 생성을 위한 위자드로 프로젝트 초기 설정과 디바이스에 간단한 설명을 명시하는 기능을 한다. 본 논문에서 지원하는 디바이스는 IVI(In-Vehicle Infotainment), 스마트폰이고 향후 웨어러블 디바이스인 스마트 워치의 개발환경을 지원할 예정이다. 또한 버튼, 리스트 등의 UI 개발 환경을 지원하기 위해 Intel의 Appframework 라이브러리[5]를 사용한다.

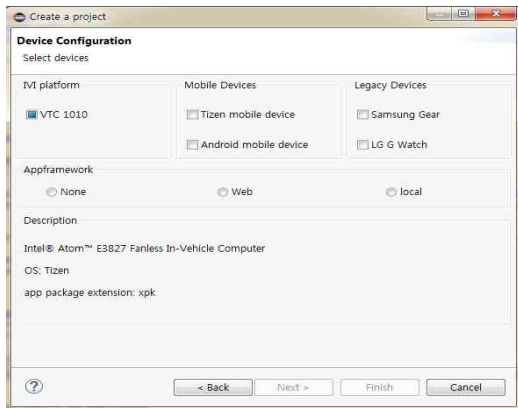


그림 2. 프로젝트 생성을 위한 위자드

그림 3은 개발환경의 진보적인 기능을 제공하는 워크벤치이다. 개발자는 HTML5, Javascript, CSS를 통해 애플리케이션을 구현할 수 있으며, 워크벤치 하단의 palette 뷰와 템플릿 코드 뷰를 통해 Appframework 관련 코드를 자동 생성하여 소스코드 에디터에 삽입할 수 있다. 또한, 프리뷰기능을 통해 사용자가 작성하고 있는 소스코드의 UI를 실시간으로 확인할 수 있다.

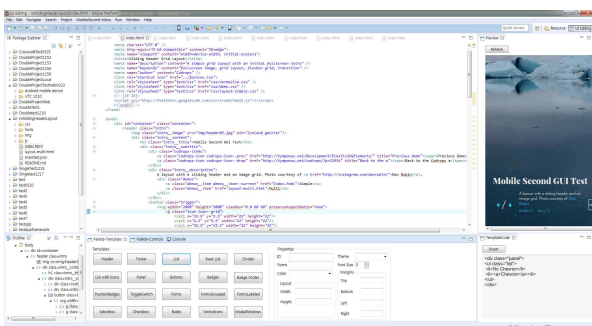


그림 3. 소스코드 작성을 위한 워크벤치

그림 4는 작성된 소스코드를 디바이스에서 실행하기위한 위자드

를 나타낸 것이다. 옵션은 총 4가지를 제공하며 각각의 옵션에 대한 설명은 다음과 같다. 첫째, xpk 패키징은 타이젠 플랫폼 디바이스를 지원하기 위한 옵션이다. 타이젠은 기본적으로 웹과 네이티브 개발 환경을 모두 지원한다. 그러나 본 논문에서는 크로스워크에서 제공하는 앱 패키징 프로세스를 기반으로 구현한다. 두 번째는 xpk를 생성하여 디바이스로 전송하는 옵션이다. IVI 디바이스 지원을 위한 옵션으로 개발자의 PC와 내부 네트워크가 연결 되어 있어야 한다는 전제조건이 있다. 3번째와 4번째 옵션은 안드로이드 디바이스를 위한 앱패키징, 전송, 실행을 위한 옵션이다.

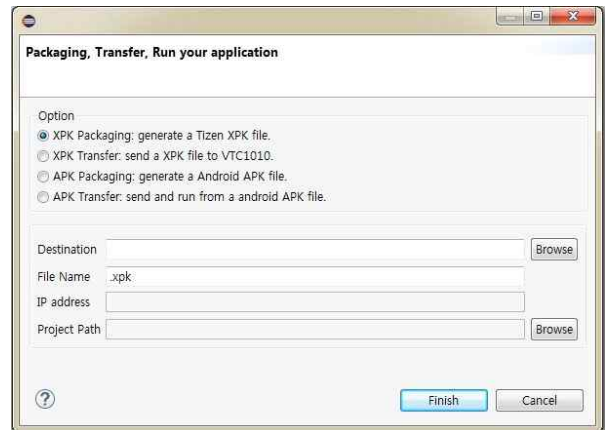


그림 4. 소스코드를 패키징, 전송, 실행하기 위한 위자드

작성된 소스코드를 테스트하는 방법은 디바이스를 이용한 방법 외에 에뮬레이터를 이용한 방법이 있다. 기존의 네이티브 언어로 작성된 소스코드는 실제 운영체제가 설치된 에뮬레이터를 사용한다. 그러나 웹 언어로 작성된 소스코드는 브라우저가 지원되는 모든 환경에서 테스트가 가능한 장점이 있다. 본 논문에서는 그림 5와 같이 브라우저 기반의 웹 에뮬레이터를 사용한다. 브라우저 기반의 웹 에뮬레이터는 구동시키는 비용이 적기 때문에 속도 측면에서 일반 에뮬레이터에 비해 빠르고 여러 개의 웹 에뮬레이터를 동시에 사용할 수 있다. 그러나 디바이스의 모든 리소스를 사용할 수 없는 단점 또한 존재한다. 이러한 단점은 웹 에뮬레이터에 가상의 플랫폼 리소스를 구현하여 극복할 수 있다. 예를 들어, 스마트폰에 장착되어 있는 가속도 센서의 경우 디바이스를 실제 움직이지 않고서는 테스트가 불가능하다. 그러나 이를 가상으로 구현하면 실제 디바이스를 사용하지 않고도 작성된 소스코드를 웹 에뮬레이터 상에서 테스트가 가능하다.

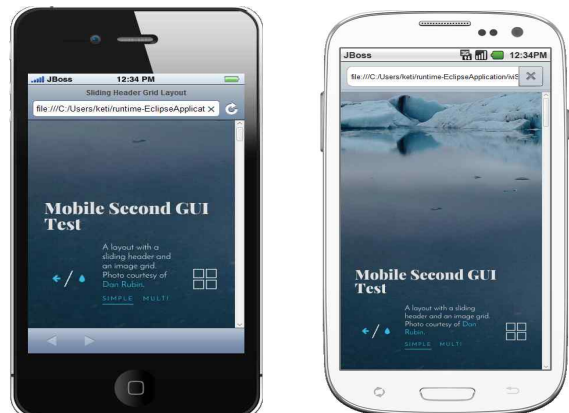


그림 5. 브라우저 기반의 웹 에뮬레이터

본 논문에서는 실제 디바이스에서 접근 가능하지만 에뮬레이터에서는 사용이 불가능한 오디오 리소스에 대한 웹 API를 설계한다. 아래의 그림 6은 크로스웍에서 제공하는 C/C++로 구현된 디바이스 리소스를 웹 언어 API로 변환한 구조를 나타낸 것이고 표1은 설계한 API에 관한 정보를 요약한 것이다.

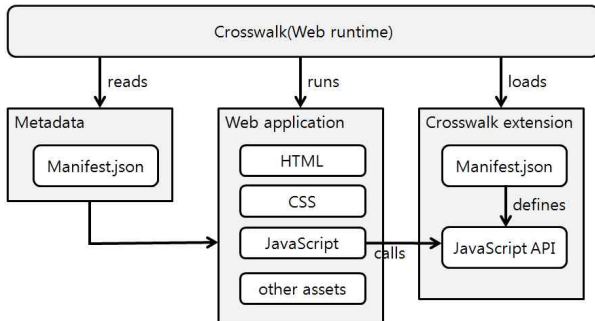


그림 6. 웹 기반 API 구현을 위한 구조



그림 7. 동일한 웹 언어로 구현된 실행 결과

표 1. 오디오 관련 주요 API 요약

API 명	Type	파라미터	설명
audioPlayer	void	none	오디오 객체 생성
Init	void	none	오디오 객체 초기설정
shutdown	void	none	Audio 객체 종료
audioStream	void	none	오디오 스트림 객체 생성
audioStart	bool	audioStream	Start 명령 전달
audioStop	bool	audioStop	Stop 명령 전달
getAudioFrame	void	sessionId	데이터 확인 메시지 전달
audioDecoder	void	None	AudioDecoder 객체 생성
dataAvailableCallback	void	sessionId, timestamp, data, len	Audio data 확인 후 받아서 저장
setupCallback	void	codecType	이벤트 값이 설정 값과 맞는지 확인
playbackStartCallback	void	sessionId	Start 명령을 Decoder에 전달
playbackStopCallback	void	sessionId	Stop 명령을 Decoder에 전달

4. 실험 및 결과

본 논문에서는 구현한 IDE의 테스트를 위하여 서로 다른 운영체제가 설치된 두 가지의 디바이스를 사용한다. 첫 번째 디바이스는 VTC 1010[6]으로 타이젠 기반의 IVI 디바이스이다. VTC 1010은 Intel® Atom 프로세서를 사용하며, GPS, WWAN, SSD 등을 지원하며, 타이젠 운영체제용 IVI profile을 포팅하여 테스트한다. 두 번째는 안드로이드 기반의 스마트폰 디바이스를 사용한다. 갤럭시 노트4를 사용한다.

그림 7은 테스트를 수행한 결과를 나타낸 것이다. VTC 1010은 공유기를 사용하여 PC와 내부 네트워크 환경을 구성하였고 스마트폰은 USB 연결을 통해 생성된 애플리케이션 설치 파일을 전송하고 실행한다. 테스트 결과 동일한 웹 언어 소스코드에 대해 서로 다른 운영체제가 설치된 디바이스에서 실행이 됨을 검증할 수 있었다.

5. 결론 및 향후 연구

본 논문에서는 멀티 플랫폼 지원을 위한 IDE를 구현하고 웹 에뮬레이터 지원을 위한 가상의 API를 설계하였다. 구현된 IDE는 다양한 디바이스에서 운영체제에 제약을 받지 않고 애플리케이션 설치가 가능하며, 웹 언어로만 작성하면 되기 때문에 앱 개발 시간을 상당히 단축시킬 수 있을 것으로 기대된다. 그리고 웹 기반의 에뮬레이터 지원을 통하여 사용자의 테스트 환경을 확장시킬 수 있었다. 그러나 웹 에뮬레이터의 경우 실제 디바이스에서 제공하는 리소스를 모두 사용할 수 없는 문제점이 있다. 이는 앞서 설계한 가상의 API를 직접 구현하여 해당 문제점을 완화할 수 있을 것이다. 또한 스마트 TV, 스마트 워치 등의 다양한 디바이스를 지원하여 시스템을 확장할 계획이다.

감사의 글

이 논문은 2014년도 정부(산업통상자원부)의 재원으로 산업융합기반 구축개발사업의 지원을 받아 수행된 연구임(No. 10037244).

참고문헌

- [1] A. Charland and B. Leroux, "Mobile Application Development: Web vs. Native," Magazine Communications of the ACM, vol. 54, no. 5, pp. 49-53, 2011.
- [2] 정구민, 강동병, 이경수, "스마트폰 크로스 플랫폼 관련 기술 개발 동향," TTA Journal vol. 144, pp. 59-64, 2012.
- [3] eclipse, <https://eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.html>
- [4] Crosswalk, <https://crosswalk-project.org/>
- [5] AppFramework, <http://app-framework-software.intel.com/>
- [6] VTC1010, <http://www.nexcom.com/Products/mobile-computing-solutions/in-vehicle-pc/in-vehicle-pc/in-vehicle-computer-vtc-1010>