

HTML5 기반 스마트 TV 플랫폼 표준 기반 TV 에뮬레이터 설계 및 구현¹

김호년, 김정현, 이동훈, 박동영
한국정보통신기술협회
{ hykimfnd, jllh, dhlee, dypark }@tta.or.kr

Design and Implementation of TV Emulator Based On HTML5 based Smart TV Platform

Ho-Youn Kim, Jung-Hyun Kim, Dong-Hoon Lee, Dong-Young Park
Telecommunications Technology Association

요 약

한국정보통신기술협회(이하, TTA)에서 제정된 “HTML5 기반 스마트 TV 플랫폼(TTAK.KO-07.0111/R1)”은 개방형 국제 기술표준인 W3C/HTML5 을 기반으로 스마트 TV 에서 방송환경과 운영체제에 종속되지 않고 애플리케이션이 실행될 수 있도록 스마트 TV 플랫폼의 기술 요구 사항을 정의한 표준이다. TTA 에서는 이러한 표준 기반의 TV 앱 생태계를 활성화하기 위해 앱 개발도구(SDK)를 개발하여 배포하였고, 이어서 TV 장치 없이 PC 환경에서 표준 기반으로 개발된 앱을 실행할 수 있는 있는 에뮬레이터 개발을 추진하였다. 개발된 에뮬레이터는 사용자의 방송 정보 설정을 바탕으로 방송 재생 및 제어 기능을 제공하고 표준 기술로 작성된 앱을 방송 연동형 혹은 패키지 형태로 실행할 수 있는 기능을 제공한다. 본 논문에서는 TTA 에서 개발한 표준 기반 TV 에뮬레이터의 설계 및 구현에 대해 소개한다.

1. 서론

“HTML5 기반 스마트 TV 플랫폼” [1][2] (이하, 스마트 TV 1.0) 표준은 IPTV · DCATV · 지상파 · 위성 등 다양한 방송 환경에서 적용 가능한 개방형 TV 플랫폼 기술 표준이다. 이 표준은 HTML5 기술을 기반으로 TV 앱 스토어를 통해 실행할 수 있는 독립형 앱뿐만 아니라 방송 연동형 서비스를 개발하고 실행할 수 있는 플랫폼 기본 기능과 멀티스크린, 적응형 스트리밍, CAS/DRM 과 같이 다양한 응용 서비스를 적용할 수 있는 고급기능을 정의하고 있다.

TTA 에서는 스마트 TV 1.0 표준 기반 서비스의 확산을 지원하기 위해 표준을 적용한 플랫폼에서 실행될 수 있는 애플리케이션을 빠르게 개발하고 검증 할 수 있는 스마트 TV 1.0 앱 개발도구(SDK) [3]를 개발하여 2014 년에 배포하였다. SDK 는 스마트 TV 표준 기반의 애플리케이션을 개발하기 위하여 프로젝트 생성, 삭제, 복사 등 프로젝트의 관리 기능을 제공하며, 표준에서 정의 하는 확장 API[4]를 문법에 맞게 사용하였는지 검증하는 기능 및 자동 완성 기능을 제공한다.

하지만, SDK 로 개발된 앱은 소프트웨어의 형식 측면에서만 검증되었기 때문에 실제 디바이스에서 실행되는 환경에서 유효성이 검증된 것은 아니다. 그리고, 개발자들에게는 표준이 적용된 디바이스를 이용하는 것도 쉽지 않은 상황이다. 삼성, LG, 구글과 같은 스마트 TV 제조사 혹은 솔루션 사에서는 고가의 디바이스 없이 PC 환경에서 개발한 앱을 실행해 볼 수 있는

에뮬레이터를 제공하고 있다.

본 논문에서는 이러한 표준기반 앱 개발 환경의 한계를 보완하여 개발자들이 실제 디바이스 없이 개발한 애플리케이션을 검증할 수 있는 스마트 TV 1.0 앱 개발용 에뮬레이터(이하, 에뮬레이터)의 설계와 구현에 대해 소개한다.

본 논문의 구성은 다음과 같다. 2 절에서는 스마트 TV 1.0 에뮬레이터의 주요 기능에 대해 설명하고, 3 절에서 에뮬레이터 설계 및 구현, 4 절에서 구현 검증을 소개한 후 마지막으로 5 절에서 본 논문의 결론 및 향후 추진 방향을 제시할 것이다.

2. 에뮬레이터 주요 기능

에뮬레이터는 실제 TV 디바이스의 동작과 동일한 사용자 경험을 제공하면서 스마트 TV 1.0 표준의 요구사항을 준수하고 앱 개발자에게 개발 중 필요한 앱 실행 정보를 제공 할 수 있도록 설계되었고 그 주요기능은 아래와 같다.

✓ 방송 정보 설정 기능

TV 의 채널, 프로그램 및 앱 시그널링 정보 등을 설정파일을 통해 구성할 수 있는 기능을 제공하며 각 채널에서 표시될 방송 영상도 함께 지정할 수 있다. 설정파일은 에뮬레이터 최상위 폴더아래 ‘si_conf’ 폴더 내 ‘service_infomatio.xml’ 이며 설정 파일의 요소 및 속성은 표 1 과 같다.

¹ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [B136-16-1002, 방송융합 기반기술 테스트 환경구축]

✓ 기본 TV 제어 기능

설정 파일의 내용을 바탕으로 설정된 각 채널에 방송 영상을 재생하고 가상 리모콘을 통해 채널 전환, 볼륨 설정 등의 기능을 제공한다. 또한, 전원버튼, 종료버튼, 방향버튼, 메뉴버튼, 칼라버튼, 숫자버튼 등 표준에서 필수 요구사항으로 정의한 키에 대응하는 리모컨 버튼 및 관련 동작을 지원한다.

표 1 애플레이터 설정 파일의 요소와 속성

요소	설명
implementation	루트 요소
services	전체 서비스의 리스트
service	서비스 정보로 name, version, id 등의 속성을 지님
source	서비스에 해당하는 가상 방송스트림
aitinfo	서비스에 속한 AIT 정보
transport	AIT의 transport 정보
label	AIT의 label 정보
baseURL	AIT의 baseURL로서 앱의 호스팅 URL의 앞부분
application	AIT의 application 정보
appid	앱의 아이디
name	앱의 이름
code	앱 제어 코드(AUTOSTART, UPDATE, KILL 등)
bound	앱의 채널 bound 여부
priority	앱의 우선순위
version	앱의 버전으로 major, minor, micro의 자식 요소
initpath	baseURL과 함께 앱의 전체 호스팅 URL을 표현함
program	해당 서비스의 프로그램 정보
id	프로그램 아이디
start	프로그램 시작 시간
dur	프로그램 전체 시간

✓ 채널 연동형 앱 실행 및 종료 기능

설정파일에 앱 시그널링 정보가 존재하는 경우 해당 채널로 진입 시 앱이 실행되고, 설정 파일에서 앱 시그널링 정보를 제거, 채널 전환, 스토어 앱 실행 등을 하는 경우에 시그널링 앱이 종료된다.

✓ 패키지 앱 실행 및 종료 기능

애플레이터 최상위 폴더아래 packages 폴더 아래 패키지 앱을 설치할 수 있으며 가상리모콘의 'menu' 버튼을 이용해 앱을 조회하고 실행 할 수 있다. 패키지 앱은 종료버튼을 인가함으로써 종료할 수 있다.

✓ 개발자 도구 기능

앱의 실행 정보(HTML 구성 정보, 리소스 정보, 콘솔, 네트워크 동작 정보) 등을 확일 할 수 있는 기능을 제공하며, 리모콘의 'info' 버튼을 통해 접근할 수 있다.

3. 애플레이터 설계 및 구현

스마트 TV 1.0 애플레이터는 실행되는 PC의 운영체제와의 종속성을 제거하기 위해 다양한 운영체제를 지원하는 웹 코어 오픈소스인 크로미엄 [5]과 자바(JAVA)로 구현되었다. 그림 1은 애플레이터의 구조를 나타낸다. CEF [6] 및 JCEF [7]는 C++ 혹은 자바로 구현된 응용프로그램이 웹코어와 통신할 수 있는 인터페이스를 제공하는 라이브러리이다. 애플레이터 MW는 TV의 기능을 지원하고 확장 인터페이스 호출에 대한 내부 처리를 수행한다. 애플레이터 UI는 애플레이터의 사용자 인터페이스 및 UI 구성을 구현하며, 애플레이터 웹 확장 인터페이스는 표준의 확장 API를 구현하는 자바스크립트 인터페이스이다.

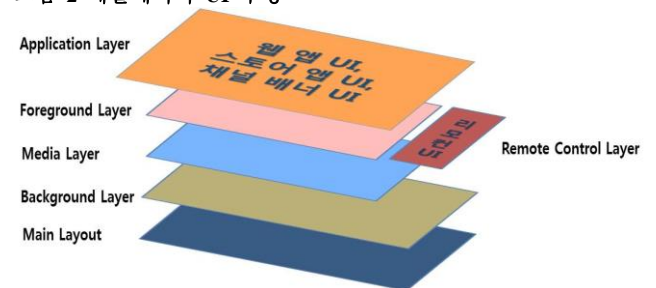
그림 1 애플레이터 구성



가. 애플레이터 UI 설계 및 구현

애플레이터 UI는 애플레이터의 사용자 인터페이스를 구현하는 부분으로 애플레이터의 UI 구성 및 가상리모콘으로 전달되는 키 이벤트의 처리 및 전달 기능을 수행한다. 애플레이터의 UI 구성은 그림 2와 같다.

그림 2 애플레이터 UI 구성



TV 패널 UI는 TV 영상을 표시하는 영역으로 Main Layout, Background Layer, Media Layer, Foreground Layer가 z-order 순으로 겹쳐지도록 구성되어 있다. Main Layout은 애플레이터의 전체 영역을 표현하고, Background Layer는 애플레이터 전체 영역의 배경색을 설정하는 영역이다. Media Layer는 방송영상이 표시되는 비디오 렌더링 영역이며, 그 윗부분을 TV의 테두리 이미지가 표시되는 Foreground Layer가 위치하게 된다. Z-order는 상위 Layer일수록 높은 값을 가지며 하위 Layer를 가리게 된다.

리모콘 UI는 그림 2에서 보는 바와 같이 Remote Control Layer에 표시되며, 리모콘 UI의 구성은 remote_control.xml의 설정에 따라 표시된다. 표 2는 리모콘

UI 를 설정하는 요소와 속성을 나타낸다. 리모콘 UI 는 키 이벤트가 발생하면 웹 앱 UI, TV 패널 UI, 스토어 앱 UI 등으로 키를 전달하는 기능을 담당한다.

웹 앱 UI 는 웹 코어에서 렌더링 되는 웹 콘텐츠를 표시하고 스토어 앱 UI 는 스토어 앱을 조회하고 실행할 수 있는 인터페이스를 제공한다. 채널 배너 UI 는 채널 전환 시 전환 된 채널의 정보를 표시한다.

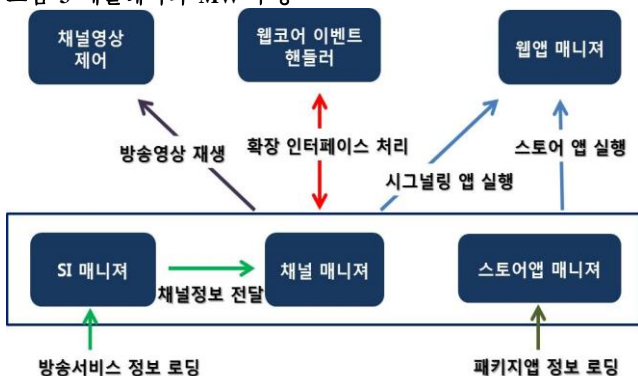
표 2 리모콘 UI 설정 파일의 요소와 속성

요소	속성	설명
remotecontrol	width	리모콘 UI의 가로크기
	height	리모콘 UI의 세로크기
	imgroot	리모콘 UI 구성 이미지들이 위치하는 폴더 위치
	backgroundimage	리모콘 UI의 백그라운드 이미지
	backgroundcolor	리모콘 UI의 백그라운드 색상
buttons	x	버튼 그룹의 x 좌표값
	y	버튼 그룹의 y 좌표값
	width	버튼 그룹 영역의 가로 크기
	height	버튼 그룹 영역의 세로 크기
button	img	버튼 이미지
	x	버튼의 좌우측 모서리의 x 좌표값
	y	버튼의 좌우측 모서리의 y 좌표값
	width	버튼의 가로크기
	height	버튼의 세로크기
	keycode	버튼 클릭 시 발생하는 키 이벤트

나. 에플레이터 MW 설계 및 구현

에플레이터 MW 는 채널 정보 및 프로그램 정보 수집 및 관리, TV 제어 API 제공, 스토어 앱 관리, 웹 앱 관리, 확장 API 처리 등을 담당하는 모듈로서 SI 매니저, 채널 매니저, 스토어앱 매니저, 웹앱 매니저, 웹코어 이벤트 핸들러, 채널영상제어 등의 하위 모듈로 구성되어 있다. 그림 3 은 에플레이터 MW 의 하위 모듈들간의 관계를 간략히 나타낸다.

그림 3 에플레이터 MW 구성



SI(Service Information) 매니저는 표 1 의 형식으로 설정된 에플레이터 설정파일로부터 서비스, 프로그램, 앱 시그널링 정보를 로딩한다. 특히, 에플레이터 실행 중 설정의 변경을 적용하기 위해 주기적으로 설정이 변경되었는지 확인하여 변경된 내용을 적용한다. 또한, 서비스 정보가 갱신될 때마다 채널 매니저에게 해당 정보를 전달한다. 채널 매니저는 SI 매니저로부터 서비스 정보를 받기 위해 SI 매니저에 이벤트 처리 함수를 등록하고, 이 함수를 통해 전달된 서비스 정보를 분석하여 채널 정보를 생성한다. 이 채널 정보에는 채널 번호 및 이름, 채널 내 프로그램 정보, 앱 시그널링 정보, 채널 영상 정보 등이 포함되며, 채널이 변경되었을 때 시그널링 정보를 웹앱 매니저에 전달하여 시그널링 앱을 실행시킨다. 또한, 스토어 앱 매니저는 패키지앱이 설치되는 위치에서 패키지앱의 설정파일로부터 패키지앱 정보를 구성하며, 스토어 앱 UI 에 이를 조회하고 실행하는 인터페이스를 제공하고,

채널영상제어는 채널 매니저로부터 해당 채널에서 재생되어야 하는 영상의 재생 정보를 전달받아 재생하고, 이를 제어하는 인터페이스를 제공한다.

웹코어 이벤트 핸들러는 애플리케이션에서 확장 API 를 호출할 때 전달되는 메시지를 분석하여 처리하고 결과를 반환해 주는 역할을 수행한다. 이때 전달되는 메시지에는 메시지 아이디와 파라미터 등이 있고, 메시지 아이디는 호출된 확장 API 를 구분할 수 있는 식별자이며 파라미터는 확장 API 의 파라미터 값들이다. 웹앱 매니저는 웹 코어를 초기화 하고 웹 확장 인터페이스의 자바스크립트 구현부를 웹 코어에 등록하며 앱을 실행하는 인터페이스를 제공한다. 또한, 웹코어에서 발생하는 이벤트를 처리할 수 있는 핸들러를 등록하는 인터페이스를 제공한다.

다. 에플레이터 웹 확장 인터페이스 설계 및 구현

웹 코어 확장 인터페이스는 표준의 확장 API 를 구현하는 모듈로 자바스크립트로 구현되었고, 자바스크립트에서 함수 호출을 자바 실행영역에서 처리하기 위해 웹코어에 웹코어 이벤트 핸들러를 등록한다.

웹코어 이벤트 핸들러의 자바스크립트 인터페이스는 아래와 같다.

- 함수명 : window.cefQuery
- 파라미터
 - ✓ request : id 와 확장 API 파라미터 문자열
 - ✓ onSuccess: 처리 완료 시 호출되는 함수
 - ✓ onFailure: 처리 실패 시 호출되는 함수

웹코어 이벤트 핸들러의 자바 인터페이스는 아래와 같다.

- 함수명 : onQuery
- 파라미터
 - ✓ request : cefQuery 의 request 문자열
 - ✓ callback.success: 처리 완료 후 호출하는 함수
 - ✓ callback.fail: 처리 실패 시 호출하는 함수

그림 4 는 자바스크립트 인터페이스와 자바 인터페이스의 구현 예를 나타낸다.

그림 4 웹 확장 인터페이스 구현 예

```

웹 확장 자바스크립트 인터페이스구현 예
self.setKeySet = function(keys){
    window.cefQuery({
        request: 'tvExt.application.appmgr.setKeyset'+keys,
        onSuccess: function(response){
            console.log('response : '+response);
            keySetValue = response;
        },
        onFailure: function(errCode,errMsg){
        });
});
};

웹 확장 자바 인터페이스 구현 예
if(request.startsWith("tvExt.application.appmgr.setKeyset")){
    System.out.println("Request : setKeyset");
    String sKeySets = request.substring(request.indexOf("setKeyset")+setKeyset.length());
    int iKeySets = Integer.parseInt(sKeySets);
    WebApp.getInstance().setKeySetN(iKeySets);
    callback.success(sKeySets);
    return true;
}
    
```

4. 구현 검증

본 논문에서 소개한 에뮬레이터는 표준기반으로 개발된 상용 서비스 애플리케이션과 표준 적합성 시험 [8]의 테스트케이스를 실행하여 검증하였다. 또한, 에뮬레이터 검증은 윈도우 7 64bit 운영체제, 8GB 메모리, Intel I7 CPU 사양의 PC를 활용하였다.

그림 5는 표준기반으로 개발된 상용 서비스 실행 예이다. 그림 6은 테스트케이스 앱의 실행화면을 나타내며 그림 7은 스토어 앱 UI의 실행 예이다.

그림 5 표준기반 상용 서비스 실행 예



그림 6 표준 적합성 시험 테스트케이스 실행 예

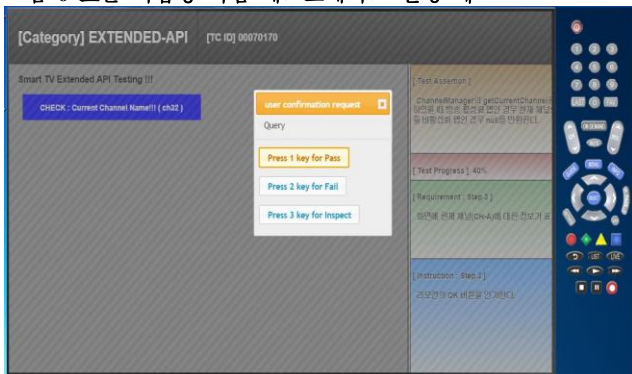
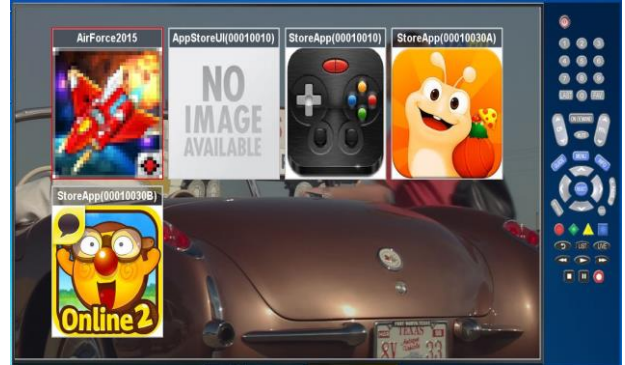


그림 7 스토어 앱 UI 실행 예



5. 결론

본 논문은 스마트 TV 1.0 표준기반으로 개발된 애플리케이션을 실제 TV 디바이스 없이 PC 환경에서 실행하고 검증할 수 있는 스마트 TV 1.0 앱 개발용 에뮬레이터의 설계와 구현에 대해 소개하였다. 에뮬레이터는 PC의 운영체제에 독립적으로 실행될 수 있도록 다양한 운영체제에서 실행되는 크로미엄 웹코어와 자바가산머신을 기반으로 개발되었고, 방송정보 설정 기능, 방송 연동형 앱 실행 기능, 패키지 앱 실행 기능, 개발자 도구 기능 등을 제공한다. 또한, 표준에서 정의한 확장 API를 구현하여 이를 활용한 애플리케이션 개발 및 검증이 가능하도록 하였다.

앞으로 TTA는 에뮬레이터의 사용자 인터페이스를 개선하고 멀티스크린, 고급 입력 인터페이스, 원격 앱 제어와 같은 고급 기능을 확대 개발할 계획이며, 이를 앱 개발사에 배포하여 상용 서비스 개발을 지원할 예정이다. 또한, 새로운 표준 기술 검증에 활용하여 표준 기술 개발도 지속적으로 추진해 나갈 것이다.

참고 문헌

- [1] TTA, "HTML5 기반 스마트 TV 플랫폼", 2014.04
- [2] 이동훈, "HTML5 기반 스마트 TV 플랫폼 표준", TTA Journal Vol.147, 2013.05
- [3] 황희선, 김호년, 이동훈, 박동영, 이은향, "HTML5 기반 스마트 TV 플랫폼 표준 앱 검증 도구 개발", 2014년 방송공학회 하계학술대회, 2014.06
- [4] 이동훈, 김호년, 이동훈, 박동영, 이은향, "HTML5 기반 스마트 TV 플랫폼을 위한 확장 API의 설계와 구현", 2013년 방송공학회 하계학술대회, 2014.06
- [5] The Chromium Projects, <https://www.chromium.org/Home>
- [6] CEF 홈페이지, <https://bitbucket.org/chromiumembedded/cef>
- [7] JCEF 홈페이지,
- [8] Hhomepage, <https://bitbucket.org/chromiumembedded/java-cef>
- [8] 이동훈, 김호년, 박동영, 이은향, "HTML5 기반 스마트 TV 플랫폼 표준에 대한 수신기 적합성 시험환경의 구현", 2014년 방송공학회 하계학술대회, 2014.06