

HTML5 기반 스마트 TV 플랫폼 표준 앱 개발도구 개발¹

황희선, 김호년, 이동훈, 박동영
한국정보통신기술협회

{ domich.hwang, hykimfnd, dhlee, dypark }@tta.or.kr

Implementation of Software Development Kit
for HTML5-based Smart TV Platform

Hee-Seon Hwang, Ho-Youn Kim, Dong-Hoon Lee, Dong-Young Park
Telecommunications Technology Association

요 약

한국정보통신기술협회(TTA)은 “HTML5 기반 스마트 TV 플랫폼(TTAK.KO-07.0111/R1)” 표준을 제정하였고, 스마트 TV 용 앱 생태계 활성화를 위해 표준기반 스마트 TV 앱 개발도구(SDK, Software Development Kit)를 개발하였다. 개발도구는 표준 기반 앱 개발의 전체 라이프사이클에 필요한 프로젝트 관리, 앱 오류검사, 애플리케이션, 패키징 및 앱 서버 전송 기능을 자동화했다. 표준에 대한 이해도가 낮은 개발자들이 표준 기반 앱을 개발 할 수 있도록 코드 자동 완성 기능과 표준 적합성 검사 기능 및 샘플 앱 등을 제공함으로써, 개발자들이 앱 자체의 기능에 대한 설계와 분석에만 집중할 수 있도록 했다. 본 논문에서는 스마트 TV 앱 개발을 위한 개발도구의 소프트웨어 구조와 개발도구가 지원하는 각 기능에 대한 구현 내용에 대해 소개한다.

1. 서론

인터넷이 고속화 되면서 데스크탑은 물론, 모바일, TV, 자동차 및 생활 가전 영역까지 웹에 접근하는 채널이 점차 다양해지고 있다.[1] 또한 모바일 디바이스가 빠른 속도로 성장하면서, 상이한 플랫폼과 OS 를 갖는 디바이스들 간의 높은 연결성과 호환성, 그리고 기기 간의 데이터 공유를 위한 처리 기술을 필요로 하는 미래형 융복합 서비스에 대한 요구가 증가하고 있다. 이에 따라, 이러한 요구사항을 실현하기 위한 앱 개발의 난이도와 복잡도도 함께 증가하고 있는 추세이다. 이런 환경에서 HTML5 는 크로스 플랫폼, 크로스 디바이스의 앱 개발을 가능하게 하여 한번의 개발로 여러 디바이스를 지원함으로써 개발의 난이도와 복잡도 문제를 해결하는 솔루션으로 급부상하고 있다. 이러한 웹 기반 서비스로의 환경 변화에 따라, 기존의 웹 기술과 기술을 통해 하드웨어를 제어하는 디바이스 API 가 융화되도록 표준화가 진행되고 있으며, TV 또한 차세대 웹 플랫폼으로 HTML5 기술을 주목하고 있다. [2]

이러한 기술 동향과 관련하여, 한국 정보 통신 기술 협회(TTA)는 IPTV, DCATV, 지상파, 위성 등 다양한 방송 환경에서 HTML5 를 기반으로 제작된 스마트 TV 용 앱이 동작하기 위한 웹 브라우저 기반의 공통 플랫폼을 정의하는 “HTML5 기반 스마트 TV 플랫폼” [3] (이하, 스마트 TV 1.0) 표준을 제정하였다. 또한 이 표준을 기반으로 적합성 시험표준을 수행하기 위해, “HTML5 기반 스마트 TV 플랫폼 수신기 적합성 시험”(이하, 스마트 TV 1.0 적합성 시험)표준[4]을 제정하였다.

본 논문은 표준 기반 앱 생태계 활성화를 위한 활동의

일환으로, 앱 개발자들이 표준 기반 앱을 쉽게 개발할 수 있도록 돕기 위해 TTA 에서 개발자들에게 제공하는 앱 개발도구에 관한 내용을 다룬다. 이 도구는 스마트 TV 앱 개발자가 TTA 에서 제정된 스마트 TV 1.0 표준에서 정의한 인터페이스를 활용하여 방송 연동형, 앱스토어형 등 다양한 스마트 TV 앱을 쉽고 빠르게 개발 및 검증할 수 있도록 돕는 개발도구이다. 스마트 TV 1.0 표준 기반의 애플리케이션을 관리하기 위한 프로젝트 생성, 삭제, 복사, импорт 등 전반적인 프로젝트 관리와 소스 코드 편집을 돕고, 표준 유효성 검증 등의 기능을 제공한다. 개발자는 이를 이용하여 표준 기반의 애플리케이션을 효율적으로 개발할 수 있다. 또한 앱 자체의 비즈니스 로직 설계 및 구현에만 집중할 수 있으며, 개발 초기 단계에 표준을 준수하는 앱을 구현할 수 있어 TV 정합 시간을 줄일 수 있다. 더 나아가 TV 앱 개발 생산성 향상 및 스마트 TV 용 앱 생태계 활성화에 기여할 수 있다.

본 논문은 다음과 같이 구성된다. 2 장에서는 표준기반 앱 개발 라이프사이클을, 3 장에서는 개발도구의 소프트웨어 구조와 확장된 기능 설계 및 구현 내용을, 4 장에서는 개발도구 기능 활용 및 결과 내용을 다룬다. 마지막으로 5 장에서는 논문의 결론을 맺는다.

2. 표준 기반 앱 개발 라이프사이클

본 장에서는 개발자가 앱 개발도구를 활용하여 앱을 개발하는 개발 라이프 사이클을 살펴본다. 그림 1 은 스마트 TV 1.0 표준을 준수하는 앱을 개발할 수 있는 개발도구와 스마트 TV 1.0 표준을 준수하는 수신기, 그리고 수신기의 표준

¹ 본 연구는 미래창조과학부의 “방송융합 기반기술 테스트 환경구축” 과제의 일환으로 수행한 결과임

적합성 기술을 검증하기 위하여 TTA 에 구축된 시험 환경 [5]이다. 앱 개발자는 이와 같은 환경에서 TTA 앱을 개발하고 검증한다.

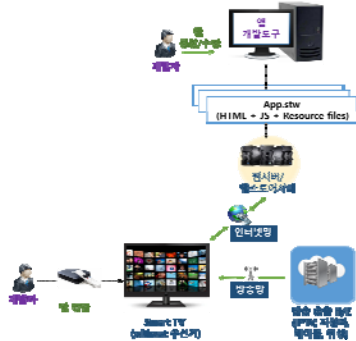


그림 1. Smart TV 앱 개발 및 정합 환경

표준 기반 앱 개발 라이프 사이클의 첫 번째, 두 번째 스텝은 개발자가 3 장에서 소개하는 개발도구의 기능들을 활용하여 앱을 개발하고 앱의 표준 적합성을 검증하는 것이다. 세 번째 스텝은 개발된 앱을 TV 에서 정합하기 위해 앱 서버에 배포하고 TV 에서 앱을 설치하여 앱이 정상 동작하는지 확인하는 것이다. 정합과정에서 문제가 발견된 경우, 개발도구를 통해 앱을 수정하여 앱 서버에 재 배포 한다. 수신기에서 스토어에 설치되었던 앱을 삭제하고 다시 설치한 후 정합 작업을 계속한다. 이와 같이 개발자는 앱을 구현, 검증하고 TV 에 정합하여 앱 서버에 배포하기까지 일련의 개발 라이프 사이클 전체에서 개발도구를 사용할 수 있다.

3. 개발 도구의 기반 환경 구조와 확장된 기능

개발도구는 이클립스라는 오픈소스를 기반으로 구현되었다. 이클립스(Eclipse) [6]는 확장 가능한 애플리케이션 프레임워크, 도구, 런타임으로 구성된 개방형 개발 플랫폼 오픈소스이다. 이 오픈소스는 다른 개발 환경에 맞춰 기능이 확장(변경)될 수 있도록 유연한 소프트웨어 구조를 갖는다. 아래에서는 확장성을 제공하는 이클립스 소프트웨어 구조에 대해 살펴보고 확장된 기능들의 구현 내용을 살펴볼 것이다.

가. 이클립스 소프트웨어 구조

이클립스는 유연한 확장성을 제공하는 소프트웨어 구조와 실행시간에 이러한 구조를 처리하여 서비스를 구동시켜주는 플랫폼 실행환경을 가지고 있다. 그림 2 는 이클립스 SDK 의 소프트웨어 구조이다. 이클립스 SDK 는 그림에서 볼 수 있듯이 이클립스 애플리케이션 개발을 위한 핵심 기반구조와 UI 프레임워크를 포함하는 RCP(Rich Client Platform), 자바 애플리케이션을 개발, 테스트 및 디버깅을 하기 위한 통합 툴들로 구성된 JD(T Java Development Tools), 플러그인을 개발, 테스트, 디버깅 및 패키징 하기 위한 통합 툴들로 구성된 PDE(Plug-in Development Environment) 컴포넌트로 구성된다. 모든 컴포넌트들은 플러그인이라는 시스템 기능 구현의 최소 단위들로 구성되어 있으며, 각 플러그인들은 확장-확장점(extension-extension point)이란 구조를 갖는다. 이러한 확장 구조는 다른 플러그인의 기능을 확장(변경)하는 구조(extension)와 자신이 확장(변경)될 수 있도록 하는 구조(extension point)를 함께 가지고 있는 것이 특징이다.

이는 플러그인이 다른 플러그인을 확장할 수 있으며, 또한 다른 플러그인에 의해 자신도 확장(변경)될 수 있도록 하기 위함이다. 플랫폼 실행환경 (Platform Runtime)은 실행시간에 구성요소들의 확장 구조 관계를 관리하며, 이클립스 서비스를 구동시키는 실행요소이다. 이것은 OSGi(Open Service Gateway initiative) [7]의 구현물로, 재시동 없이 플러그인들을 동적으로 설치, 실행, 중단, 업데이트, 제거하며 이클립스 서비스를 구동시킨다.

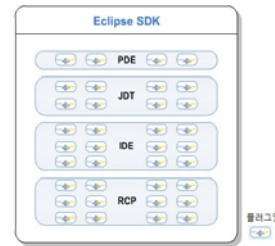


그림 2. 이클립스 소프트웨어 구조

이러한 이클립스의 소프트웨어 구조는 소프트웨어 복잡도를 크게 증가시키지 않으면서 개발도구의 추가적인 확장 없이 모든 유형의 기능 확장을 가능하게 한다. 또한 가능한 공통의 리소스나 배포 메커니즘을 활용하게 되므로 전체적인 복잡도는 오히려 감소한다. 이런 플러그인 아키텍처 특성으로 이클립스 플랫폼은 여러 회사에서 회사 제품 개발 환경에 맞게 확장(변경) 되어 사용되고 있다.

나. 프로젝트 관리

프로젝트 관리 기능은 Smart TV 앱 프로젝트를 생성해주는 기능이다. 개발자로부터 표 1 에서 명시한 프로젝트, 어플리케이션 정보를 입력 받아 그림 3 과 같이 어플리케이션의 기본 구성 파일을 생성해준다.

| 분류 | 항목 | 설명 |
|--------|-----------------------------|---|
| 프로젝트 | Project Name | 프로젝트의 이름 |
| | Organization ID | 기관의 ID |
| | Application ID | 애플리케이션 ID |
| | Application Name | 애플리케이션 이름 |
| | Application Version | 애플리케이션 버전 |
| 애플리케이션 | Resolution | 화면의 해상도 |
| | Profile | 프로파일 정보로 애플리케이션이 실행될 수신기의 표준의 구현 범위 및 서비스 특성 |
| | Type | 애플리케이션의 시작 시, 방송 활성화/비활성 여부 |
| | Offline | 애플리케이션이 네트워크에 연결이 되지 않는 상태에서 실행 가능한지 여부를 나타내는 값 |
| | Domain | 애플리케이션이 실행되는 도메인 |
| | Resource Access Policy Rule | 리소스에 대한 접근 권한 설정 (Permit) 해당자원에 대한 접근승인 톨 선택 |

표 1. 어플리케이션 설정 내용

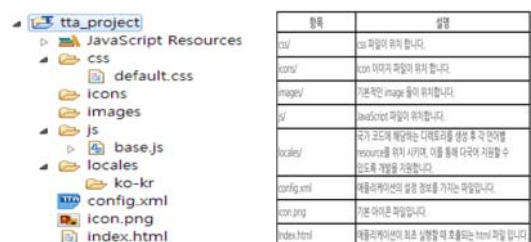


그림 3. 어플리케이션 기본 구성 파일

각 구성 파일은 그림 3 의 오른쪽 그림과 같은 역할을 한다. 구성파일 중 config.xml 파일은 개발자의 입력 정보를 기반으로 만들어지는 앱의 설정 파일(config.xml)로 수신기가

애플리케이션을 실행시킬 때 어플리케이션 정보를 얻기 위해 참조한다.

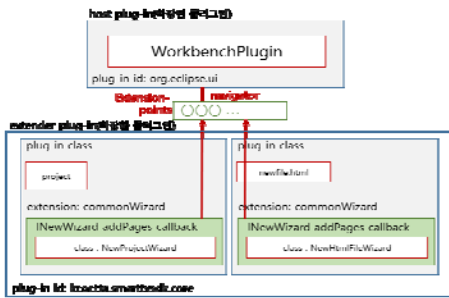


그림 4. 프로젝트 관리 기능 확장 구조 설계

프로젝트 관리 기능을 구현하기 위한 확장 구조 설계 사항은 그림 4 와 같다. 이클립스의 Workbench[8] 플러그인의 서브패키지인 navigator 에 정의된 INewWizard 인터페이스를 구현하였으며, 상세 구현 내용은 표 2 에서 설명한다.

| 기능 | Class 이름 | 설명 |
|----------------|-------------------|--------------------------------------|
| 프로젝트정보 입력 UI | NewProjectWizard | 프로젝트 정보를 입력 받고, 입력받은 정보의 유효성을 검증한다 |
| 애플리케이션 정보입력 UI | NewHtmlFileWizard | 어플리케이션 정보를 입력 받고, 입력받은 정보의 유효성을 검증한다 |

표 2. 확장한 플러그인 구현 상세 내용

다. 코드 편집

코드 편집 기능은 스마트 TV 앱의 소스코드를 작성할 수 있는 기본 기능이다. 표준 API 목록과 설명을 보여주고 API 에 대한 코드 자동완성과 오류분석 기능을 제공한다.

코드 자동완성 기능은 개발자가 API 의 첫 글자를 쓰면, 그 글자로 시작하는 API 목록을 보여주는 기능으로, JSDT 플러그인[9]을 확장하여 구현하였다. 그림 5 와 같이 JSDT API 정의 포맷으로 TTA 확장 API 정보를 구현하였고 JSDT 는 일반적인 Java Script API 에 확장된 API 를 추가 적용하여 코드 자동 완성 기능을 지원한다. 이 기능은 개발자가 프로젝트 생성시 입력하는 수신기의 프로파일과 옵션 기능 관련 API 만 보여주도록 구현하였다. TV 에서 지원하지 않는 기능을 사용하는 실수를 방지하기 위함이다.

```

1  /**
2   * SmartTVApp
3   * @returns (SmartTVApp)
4   * @memberof SmartTVExt
5   */
6  SmartTVExt.prototype.application = new SmartTVApp();
7  /**
8   * Application API
9   */
10 function SmartTVApp() {
11 /**
12  * application manager
13  */
14 SmartTVApp.prototype.appmgr = new ApplicationManager();
15 /**
16  * store app information
17  */
18 SmartTVApp.prototype.storeApp = new StoreAppInfo();
19 /**
20  * ApplicationManager API
21  */
22 function ApplicationManager() {
23 ApplicationManager.KEY_RED = 1;
24 ApplicationManager.KEY_GREEN = 1;
25 ApplicationManager.prototype.visible = true;
26 ApplicationManager.prototype.keySetValue = 1;
27 ApplicationManager.prototype.permission = 1;
28 /**
29  * set keyset
30  * @param keys
31  */
32 ApplicationManager.prototype.keySet = function(keys) {
33 /**
34  * show application
35  */
36 ApplicationManager.prototype.showApplication = function() {

```

그림 5. JSDT API 정보 파일

오류분석 기능은 개발 언어에 따라 두 가지 방식으로 구현하였다. 첫 번째 방식으로 HTML 이나 CSS 코드의 경우 개발자가 코드 편집하는 동안 자동으로 수행되는

이클립스의 Validation 기능을 확장하여 구현하였다. Validation 기능은 개발자가 업데이트한 코드를 콜백 함수를 통해 플러그인에 전달해 주며 플러그인은 API 의 물을 명시한 파일을 읽어 업데이트된 코드 패턴을 찾아 허용되는지 검사한 후 안되는 경우 에러 마크를 표시해 준다. 두 번째 방식으로 JS 코드의 경우 앱 패키지 생성 후 개발자가 이클립스의 커맨드 버튼을 눌렀을 때 수행된다. 패키지 파일을 입력으로 표준 적합성 검사를 하고 오류 결과를 리포트해주는 외부 검증도구[10]를 실행시켜 받은 OUTPUT 파일 내용을 화면에 표시한다.

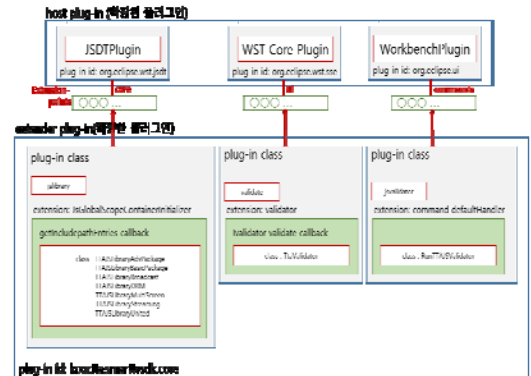


그림 6. 코드 자동완성과 오류검증 기능 확장 구조 설계

코드자동완성과 오류검정 기능을 구현하기 위한 확장 구조 설계 사항은 그림 6 과 같으며, 이클립스의 JSDT, WST Core, Workbench 플러그인의 core, ui, commands 서브 패키지 기능을 확장하여 구현하였다. 상세 구현 내용은 표 3 에서 설명한다.

| 기능 | Class 이름 | 설명 |
|----------|------------------------------|---|
| 코드자동완성 | TTAJSLibraryAdvPackage | 수신기 프로파일과 옵션기능에 따른 확장 API Set 을 JSDT 플러그인에 등록한다 |
| | TTAJSLibraryBasicPackage | |
| | TTAJSLibraryBroadcastPackage | |
| | TTAJSLibraryDRM | |
| | TTAJSLibraryMultiscreen | |
| | TTAJSLibraryUnited | |
| 웹 코드 검증 | TTAValidator | HTML, CSS 코드의 오류 분석 기능을 구현한 Class 로, 사용이 제한된 API 가 사용되었는지 검사하고 오류결과를 개발자에게 보여준다 |
| JS 코드 검증 | RunTTAJSValidator | JS 코드의 오류분석 기능을 구현한 Class 로, 외부검증도구를 실행시켜 오류 결과를 받아와 개발자에게 보여준다 |

표 3. 확장한 플러그인 구현 상세 내용

라. 앱(App) 패키징, 애플리케이션 및 앱 서버 업로드

앱 패키징 기능은 개발된 스마트 TV 표준 앱을 W3C widget 표준에서 정의하는 패키징 방식에 맞춰 패키징하는 기능이다. 애플리케이션 기능은 개발된 앱을 PC 브라우저에서 애플리케이션 하는 기능으로, PC 에 설정된 HTML 파일에 대한 연결프로그램을 구동시켜 앱을 실행시킨다. 업로드 기능은 패키징된 앱을 앱 서버에 업로드 하는 기능으로, 개발자가 미리 설정한 앱 서버의 주소와 위치로 패키징 된 앱을 전송한다.

각 기능을 구현하기 위한 확장 구조 설계 사항은 그림 7 과 같으며, Wokbench 플러그인의 서브패키지 commands 에서 정의한 IObjectActionDelegate 인터페이스를 구현하였으며, 상세 구현 내용은 표 4 에서 설명한다.



그림 7. 앱 패키징, 애플리케이션, 업로드 기능 확장 구조 설계

| 기능 | Class 이름 | 설명 |
|--------|---------------------|-----------------------------------|
| 패키징 | BuildSTWidgetAction | 앱을 W3C Widget 표준에서 정의한 방식으로 패키징한다 |
| 앱서버 전송 | DeploySTWAction | 패키징된 앱을 앱서버에 전송한다 |
| 애플리케이션 | RunEmulationAction | 앱을 PC 브라우저에서 애플리케이션한다 |

표 4. 확장한 플러그인 구현 상세 내용

마. 개발도구 업데이트

개발도구 업데이트 기능은 이클립스 런칭 시 개발 도구 소프트웨어 버전 관리 서버에 연결하여 업데이트된 버전이 있는 경우 개발자에게 업데이트를 권고하여 개발도구를 업데이트 할 수 있도록 하는 기능이다.

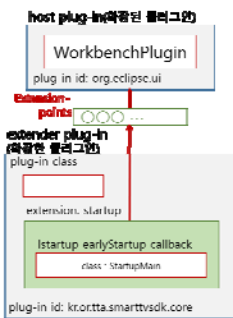


그림 8. 개발도구 SW에 대한 배포 및 갱신 기능 확장 구조 설계

업데이트 기능을 구현하기 위한 확장 구조 설계 사항은 그림 8 과 같으며, Wokbench 플러그인의 서브패키지 ui 에서 정의한 IStartup 인터페이스를 구현하였으며, 상세 구현 내용은 표 5 에서 설명한다.

| 기능 | Class 이름 | 설명 |
|------|-------------|---|
| 업데이트 | StartupMain | 서버에 최신 릴리즈된 버전이 있는 경우 새 버전 설치를 권고하는 UI 를 개발자에게 보여준 후 업데이트를 진행한다 |

표 5. 확장한 플러그인 구현 상세 내용

4. 개발도구 기능 활용 및 결과

본 논문에서 소개한 앱 개발도구는 참조 수신기에서 실행될 애플리케이션을 개발하는데 활용되었다. 그림 9 는 앱 개발도구를 활용하여 게임 앱을 개발하고 앱서버에 업로드하여 TV 에 정합하는 과정을 나타낸다. 왼쪽 상단은 코드 에디팅 화면, 오른쪽 상단은 애플리케이션 화면, 왼쪽 하단은 앱서버를 통한 애플리케이션 설치 화면, 오른쪽 하단은 앱 실행 및 정합 화면이다.

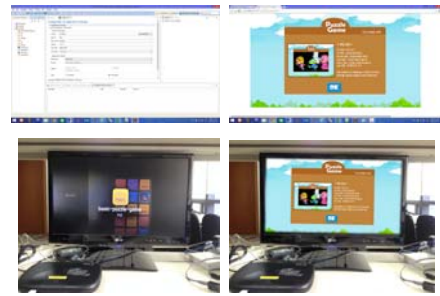


그림 9. 개발도구 사용 사례

앱 개발도구를 활용함으로써 프로젝트 생성, 앱 코딩 및 오류검사, 애플리케이션, 패키징 및 앱 서버 전송 등 개발 라이프 사이클 전체에서 도구가 사용됨을 확인할 수 있었다.

5. 결론

본 논문은 TV 용 앱 생태계 활성화를 위해 앱 개발자에게 제공되는 표준 기반 스마트 TV 앱 개발도구의 설계와 구현에 대한 내용을 다루었다. 스마트 TV 용 표준 앱을 효율적으로 개발하기 위한 프로젝트 관리, 코드 에디터, 코드 개발 지원도구, 유효성 검증 도구, 배포 도구 등 다양한 기능을 구현하였다. 일련의 개발 라이프 사이클 전체에서 이 개발도구가 사용되어 앱 개발 생산성이 향상되길 기대한다. 또한 이 도구가 이클립스 확장성 구조를 살려, HTML5 기반으로 앱이 작성되는 다양한 디바이스의 앱 개발 도구로 확장되어 사용되어지길 기대한다.

현재 도구의 애플리케이션 애플리케이션 기능은 일반 웹 기능에 대해서만 지원된다. 앞으로 TTA 에서는 확장 API 까지 애플리케이션 가능하도록 이 기능을 업그레이드 할 계획이다.

참조문헌

[1] Appcelerator, IDC, 2015 년 사용 예상 기기, 2013

[2] 박세환, 김정호, 유대상, “ HTML5 기반 스마트 TV 플랫폼 산업 동향 ”, 정보통신산업진흥원, 2013.7.3

[3] TTA.KO.07-0111/R1, “ HTML5 기반 스마트 TV 플랫폼 ”, 2014.04

[4] TTA.KO.07-0119/R1, “ HTML5 기반 스마트 TV 플랫폼 수신기 표준 적합성 시험 ”, 2015.04

[5] 이동훈, 김호년, 박동영, 이은향, “ HTML5 기반 스마트 TV 플랫폼 표준에 대한 수신기 적합성 시험환경의 구현 ”, 2014 년 방송공학회 하계학술대회, 2014.06

[6] Eclipse Homepage, <http://www.eclipse.org>

[7] OSGi Alliance Homepage, <http://www.osgi.org/Main/HomePage>

[8] Eclipse Workbench Plugin, <https://eclipse.org/articles/Article-WorkbenchSelections/article.html>

[9] Eclipse JSDT Plugin, <https://eclipse.org/webtools/jsdt/>

[10] 황희선, 이동훈, 김호년, 박동영, 이은향, “ HTML5 기반 스마트 TV 플랫폼 표준 앱 검증도구 개발 ”, 2014 년 방송공학회 하계학술대회, 2014.06

[11] Eclipse WST Core Plugin, <https://eclipse.org/webtools/wst/components/sse/archiveOfOldish/PluginOverview.html>