

HTML5 기반 스마트 TV 플랫폼 표준 적합성 시험을 위한 스마트 TV 테스트 프레임워크 설계와 구현¹

김호년, 이동훈, 박동영, 이은향
한국정보통신기술협회
{ hykimfnd, dhlee, dypark, ehlee }@tta.or.kr

Design and Implementation of Smart TV Test Framework for Conformance Test Environment of HTML5 based Smart TV Platform

Ho-Youn Kim, Dong-Hoon Lee, Dong-Young Park, Eun-Hyang Lee
Telecommunications Technology Association

요 약

본 논문은 한국정보통신기술협회(TTA)에서 제정된 “HTML5 기반 스마트 TV 플랫폼 수신기 적합성 시험” 표준(TTAK.KO-07.0119)에 따라 구축한 표준 적합성 시험환경에서 시험 도구의 테스트 케이스 개발에 활용되는 스마트 TV 테스트 프레임워크 설계와 구현에 대한 내용을 소개한다. 표준으로부터 도출된 많은 개수의 테스트 항목에 대한 테스트 케이스를 효율적으로 개발하기 위해 테스트 내용 외의 공통적으로 수행되는 코드 부분(테스트 케이스의 정의, 기본 UI, 서버와의 통신 등)을 추상화하고, 테스트 케이스 개발 자체의 난이도를 줄이기 위해 사용하기 쉬운 API 를 제공하는 프레임워크가 필요하다. 본 논문에서는 스마트 TV 테스트 프레임워크의 소프트웨어 구조를 소개하고, 테스트 케이스 개발에 활용하는 테스트 서술 API, 테스트 결과를 사용자에게 일관되게 보여주는 테스트 프레임워크 UI, 그리고 테스트 결과를 수집하고 서버에 전송하는 테스트 리포터의 설계와 구현 내용에 대해 소개한다.

1. 서론

HTML5/W3C, HbbTV, HTML5 기반 스마트 TV 플랫폼^[1] 등과 같은 플랫폼 표준이 개발되면 구현 플랫폼의 표준 적합성을 시험하기 위해 테스트 케이스를 개발한다. 이러한 테스트 케이스 개발 과정에서 관련 표준 기술이 지속적으로 발전함에 따라 추가로 개발해야 할 테스트 케이스가 많아진다. 따라서, 효율적으로 많은 양의 테스트 케이스를 개발하기 위해서는 테스트 케이스 개발을 위한 프레임워크가 필요하다. 특히, 다양한 개발자가 함께 개발에 참여할 때 프레임워크를 통해 개발함으로써 같은 기능을 각자가 개발해야 하는 부담을 줄일 수 있을 뿐만 아니라, 쉽게 테스트 케이스 개발에 참여할 수 있다. W3C 에서도 테스트 케이스 개발을 지원하기 위해 testharness.js^[2]를 개발자들에게 배포하고 있다.

본 논문에서는 “HTML5 기반 스마트 TV 플랫폼” 표준(이하, 스마트 TV 표준)에 따라 구현된 수신기의 표준적합성 시험을 위하여 테스트 케이스 개발에 활용되는 스마트 TV 테스트 프레임워크(이하, 테스트 프레임워크)를 설계하고 구현한 내용에 대해 소개하고, 이를 기반으로 개발된 테스트 케이스들을 참조 수신기의 표준적합성 시험에 적용한 내용을 다룬다.

본 논문의 구성은 다음과 같다. 2 절에서 TTA 에서 구축한 표준적합성 시험환경(이하, 스마트 TV 적합성 시험환경)에 대해 소개하고, 3 절에서는 스마트 TV 테스트 프레임워크의 설계와 구현에 대해 소개한 후 4 절에서 스마트 TV 표준 적합성 시험 적용 및 검증에 대해 다루고, 마지막으로 5 절에서는 본 논문에 대한 결론을 맺는다.

2. 스마트 TV 적합성 시험환경

2013 년 3 월 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” 표준이 TTA 에서 제정되었다. 또한, 위의 표준에 따라 개발된 수신기가 표준의 기술 요구사항을 준수하는지 검증하기 위해 시험 기준과 수행절차, 판정 기준 등을 제시하는 “HTML5 기반 스마트 TV 플랫폼 수신기 적합성 시험” 표준^[3](이하, 스마트 TV 적합성 시험 표준)이 제정되었다.

이러한 기술표준과 관련하여, TTA 에서는 스마트 TV 적합성 시험 표준에 따라서 스마트 TV 적합성 시험환경을 아래 그림 1 과 같이 구축하였다. 시험환경에서 실행되는 각 테스트 케이스는 테스트 프레임워크 기반으로 웹 앱(Web

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

App)의 형태로 개발되었다. 테스트 케이스는 시험대상장비에 설치되어 실행되고 테스트 결과를 앱 UI 와 표준 적합성 관리 시스템에 전달하게 된다. 개발된 테스트 케이스는 형상관리 서버에서 SW 의 형상이 관리되며, 웹 서버와 표준 적합성 관리 시스템에 동기화 서비스를 제공한다.

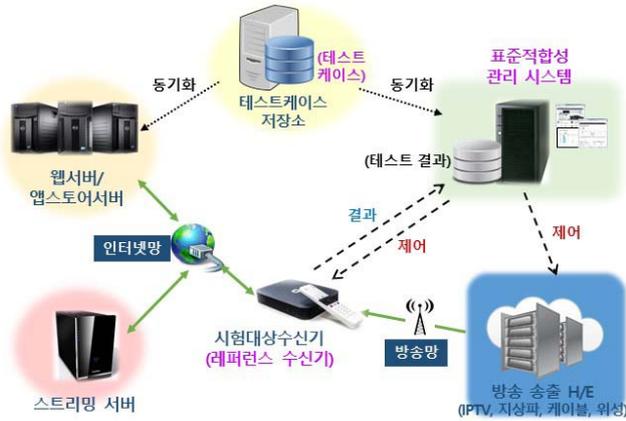


그림 1. 스마트 TV 수신기 적합성 시험환경의 구성

3. 스마트 TV 테스트 프레임워크 설계 및 구현

그림 2 는 테스트 프레임워크의 구성을 나타낸다. 테스트 프레임워크는 자바스크립트 단위 테스트에 사용되는 Jasmine library^[4]와 jQuery 그리고 jQuery UI^[5]를 기반으로 한다. Jasmine library 은 MIT license 의 오픈 소스로서 테스트 과정을 정의할 수 있는 API, 다양한 결과 판정 API, 비동기 테스트 그리고 외부 모듈과 통신할 수 있는 API 등의 기능 등을 제공한다. jQuery UI 및 jQuery library 는 팝업창, 메뉴, 버튼, 탭 등 다양한 UI 요소를 쉽게 생성할 수 있는 API 와 html tag 를 쉽게 제어할 수 있는 API 를 제공한다.

테스트 프레임워크는 이와 같은 기반 library 를 바탕으로 테스트 서술 API, 테스트 프레임워크 UI 그리고, 테스트 리포터 등의 모듈로 구성되어 있다. 테스트 서술 API 는 테스트 케이스와 수행할 테스트 내용을 정의하기 위한 API 이고, 프레임워크 UI 는 테스트 결과를 일관되게 보여주기 위해 html tag 를 미리 정의하고 각 UI 요소의 디자인을 정의하는 CSS 로 이루어져 있다. 테스트 리포터는 Jasmine 에서 테스트 결과를 전달받아 프레임워크 UI 를 업데이트하고 연결된 시험 결과 수집 서버에 정보를 전달하는 모듈이다.



그림 2. 테스트 프레임워크 구성

가. 테스트 서술 API 설계 및 구현

TTA 에서는 시험항목을 정의하기 위해 시험의 목적, 절차, 판정기준 등의 정보를 XML 포맷으로 작성하는 테스트 기술서(test description)를 개발하였다. 테스트 기술서는 그림 3 과 같은 테스트 기술 XML 스키마를 기반으로 작성된다.

테스트 서술 API 는 테스트 기술서와 대응되는 형태로 테스트 케이스를 개발 할 수 있도록 테스트 기술 XML 스키마를 바탕으로 설계되었다.

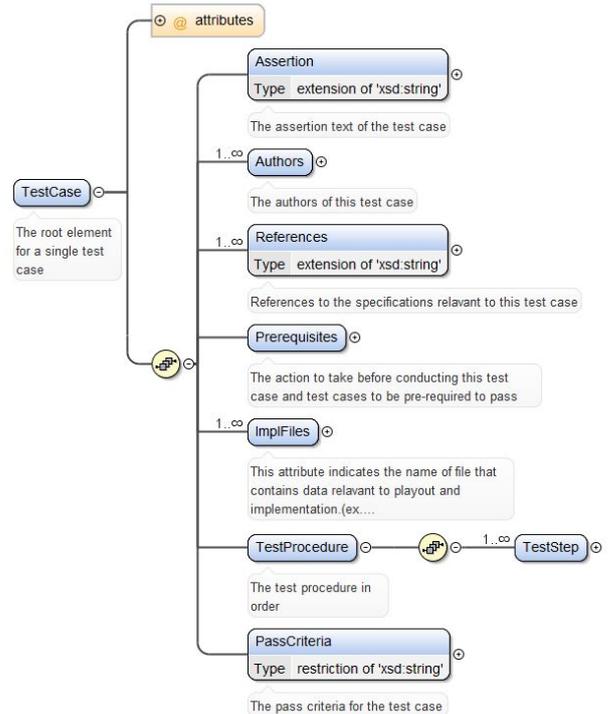


그림 3. 테스트 기술 XML 스키마

표 1 은 테스트 서술 API 의 리스트를 나타낸다. TestCase 함수는 테스트 기술 XML 스키마의 TestCase 요소에 대응하는 API 로써 Jasmine engine 에 각 단계별 테스트 실행 코드를 등록하는 역할을 한다. TestStep 함수는 테스트 기술 XML 스키마의 TestStep 요소에 대응하는 API 로써 단계별 테스트 코드를 정의한다. 테스트 케이스 앱이 실행되면 테스트 프레임워크는 기반 library 들을 동적으로 로딩하고, 테스트 기술서 파일로부터 테스트 정보를 추출한 이후 Jasmine engine 을 시작 시키고, 등록된 TestStep 의 각 함수들은 Jasmine engine 에 의해 순차적으로 실행된다.

requestUserAction 함수는 시험원에게 특정 입력(채널 전환, 특정 키 입력 등)을 유도하는 팝업창을 표시하고, askForConfirmation 함수는 이미지 비교와 같이 프로그램에 의해 자동으로 결과를 판단할 수 없을 때 사용자로부터 테스트 결과를 직접 입력하도록 하는 팝업창을 표시한다. 이 두 함수는 jQuery UI 의 다이얼로그 생성 기능을 사용한다.

checkTestStepStatus 함수는 tId 가 지정하는 테스트 스텝의 실행 유/무와 그 결과 등 정보를 조회하는 함수이다. 여러 테스트 케이스 앱을 통하여 하나의 테스트 케이스가 실행되거나 하나의 테스트 케이스 앱이 종료되었다가 다시 실행되는 경우 다른 앱에서 실행된 테스트 스텝의 실행상태를 확인할 수 있는 기능을 제공한다.

이러한 테스트 서술 API 들로 작성된 테스트 케이스는 그림 4 와 같은 순서로 실행된다.

함수명	함수인자
TestCase	<ul style="list-style-type: none"> • tId: 테스트 케이스 아이디 • func: 실행할 테스트 케이스 함수
TestStep	<ul style="list-style-type: none"> • nStep: 테스트 스텝 아이디 • func: 실행할 테스트 스텝 함수
askForConfirmation	<ul style="list-style-type: none"> • queryMsg: 팝업창에 표시할 문구 • preProcess: 팝업창 표시 전 실행할 함수 • postProcess: 팝업창에서 확인 혹은 취소를 선택한 이 후 실행될 함수 • timeout: 사용자 반응을 기다리는 최대 시간(ms)
requestUserAction	<ul style="list-style-type: none"> • queryMsg: 팝업창에 표시할 문구 • preProcess: 팝업창 표시 전 실행할 함수 • postProcess: 팝업창에서 pass, fail 혹은 inspect 를 선택한 이 후 실행될 함수 • timeout: 사용자 반응을 기다리는 최대 시간(ms)
checkTestStepStatus	<ul style="list-style-type: none"> • tId: 테스트 케이스 아이디 • nStep: 테스트 스텝 아이디

표 1. 테스트 서술 API 리스트

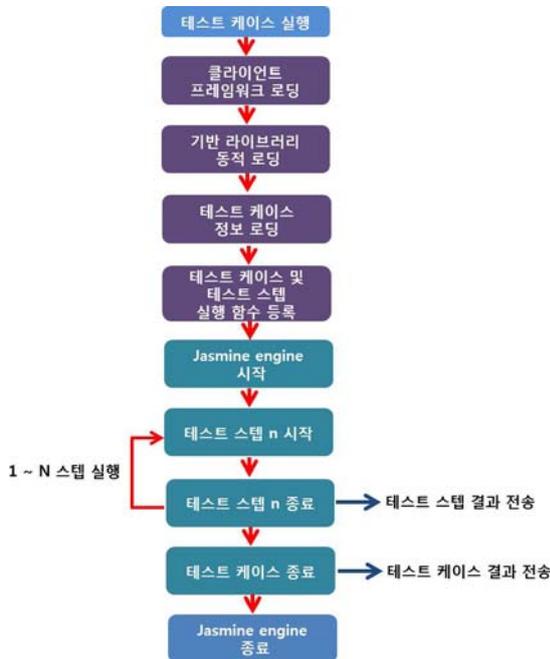


그림 4. 테스트 케이스 앱 실행 순서도

나. 테스트 프레임워크 UI

본 테스트 프레임워크는 테스트 진행 과정과 결과를 일관되게 표시하고, 시험 진행과정과 결과를 쉽게 파악할 수 있도록 하며, 테스트 케이스 개발자가 UI/UX 디자인 개발에 대한 부담을 덜 수 있도록 프레임워크 수준에서 UI 를 제공한다.

아래 그림 5 는 테스트 프레임워크 UI 의 구성을 나타낸다. ① ~ ⑥ 은 UI 를 구성하는 div 요소들이고, 테스트 케이스 앱

초기화 시 동적으로 생성되며, 각 요소들의 상세 디자인은 별도의 파일에 CSS 로 명시되어 있다. tc_header 요소 영역은 테스트 케이스의 카테고리 및 아이디 정보를 표시하고, test description 으로부터 정보를 조회하여 표시된다. app_area 요소 영역은 테스트 케이스가 필요한 커스텀 UI 요소가 표시되는 영역으로써, 개발자는 app_area 영역 내에 필요한 요소를 선언하여 사용하여야 한다. assertion 요소 영역은 test description 의 assertion 의 내용을 표시한다. tc_progress 요서 영역에는 테스트의 진행율이 표시된다. test_req 요소 영역에는 현재 진행 중인 테스트 스텝의 판정 기준이 표시되고, next_desc 요소 영역에는 다음 스텝의 설명이 표시된다. 모든 테스트가 완료되면, 최종결과는 next_desc 영역에 표시된다.

테스트 프레임워크 UI 는 1280x720 과 960x540 해상도를 지원하며, 테스트 케이스 실행 전 API 호출을 통해 어떤 UI 로 표시할 지 설정할 수 있다. 기본 설정은 1280x720 이다.

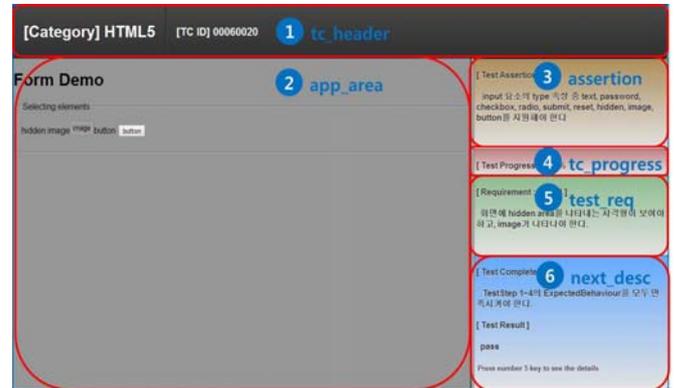


그림 5. 테스트 프레임워크 UI 구성

다. 테스트 리포터 설계 및 구현

Jasmine library 는 테스트가 진행되는 과정에서 reporter 라는 객체의 멤버함수의 호출을 통해 외부 모듈과 통신을 한다. 함수를 호출하게 되는 이벤트는 테스트케이스시작, 테스트케이스종료, 테스트스텝시작, 테스트스텝종료 등이 있고, 종료 이벤트 발생 시 테스트 결과를 reporter 에 전달한다. Reporter 객체는 Jasmine library 의 API 를 통해 Jasmine engine 에 등록 할 수 있다. 본 논문에서 구현한 reporter 객체인 Test reporter 에서는 각 함수가 호출될 때 연관된 테스트 프레임워크 UI 의 요소를 이벤트에 맞게 업데이트하고, 테스트 케이스 전체 결과 및 테스트 스텝의 결과들을 연결된 테스트 결과 수집 서버에 전송하는 역할을 한다. 아래 표 2 는 reporter 객체의 함수 리스트를 나타낸다.

함수명	설명
reportRunnerStarting	테스트 케이스의 실행이 시작될 때 호출됨
reportRunnerResults	Jasmine engine 이 종료될 때 호출됨
reportSuiteResults	테스트 케이스의 실행이 종료될 때 호출 됨 (테스트 케이스 전체 결과 전달)
reportSpecStarting	테스트 스텝이 실행될 때 호출 됨
reportSpecResults	테스트 스텝의 실행이 완료될 때 호출됨 (테스트 스텝 결과 전달)

표 2. 테스트 리포터 객체 함수 리스트

테스트 리포터는 테스트 결과 수집 서버에 결과 정보를 전달하기 위해 ajax 를 사용하며 다음과 같은 정보를 JSON 형태로 전달한다.

요소	값	설명
tcId	integer	테스트 케이스 아이디
step	integer	테스트 스텝 인덱스
pass	string	테스트 스텝 실행 결과
description	string	로그 정보

표 3. 테스트 결과 수집 서버 송신 메시지 포맷

4. 스마트 TV 표준 적합성 시험 적용 및 검증

테스트 프레임워크는 스마트 TV 적합성 시험환경의 테스트 케이스 개발에 사용되었고, 표준 레퍼런스 수신기 시험에 활용되었다. 개발된 테스트 케이스는 총 200 여개이고, 카테고리 별 시험내용과 항목 수는 아래 표 4 과 같다.

아래 그림 6 은 시험환경에서의 테스트 케이스를 실행한 예이다. 적합성 시험 관리시스템은 테스트 케이스 실행을 위해 방송송출서버에 테스트 케이스 앱 시그널링을 요청하고, 테스트 결과를 수집하는 시스템으로 WEB UI 를 통해 제어한다.

구분	시험내용	항목수
앱 실행제어	<ul style="list-style-type: none"> App Type 에 따른 수신기 동작 App 우선순위에 따른 제어 동작 	12
앱 시그널링	<ul style="list-style-type: none"> App 시그널링에 따른 생명주기 제어 AIT 각 field 속성에 따른 수신기 동작 	19
앱 패키징	<ul style="list-style-type: none"> App 패키징 포맷 및 설정파일에 따른 동작 기본 승인 정책에 따른 수신기 동작 	24
콘텐츠 포맷	<ul style="list-style-type: none"> 다양한 콘텐츠 포맷에 대한 수신기 지원 	29
프로토콜	<ul style="list-style-type: none"> RTSP/HTTP 프로토콜 지원 Widget 관련 MIME 타입 및 다운로드 지원 	4
HTML5	<ul style="list-style-type: none"> HTML5 주요 특성에 대한 수신기 동작 <video> 요소를 이용한 콘텐츠 스트리밍 	57
확장 API	<ul style="list-style-type: none"> 수신기 확장 API 의 지원 여부 각 API 에 대한 수신기 동작 	43
보안	<ul style="list-style-type: none"> 보안관련 수신기 요구 사항 보안 예외 사항 처리 	3
수신기 요구사항	<ul style="list-style-type: none"> 표준에 대한 수신기 요구 사항 KEY 입력 및 내비게이션 매커니즘 지원 	14
총 합계		200

표 4. 테스트 케이스 개발 현황



그림 6. 테스트 케이스 실행 예

5. 결론

본 논문에서는 스마트 TV 적합성 시험 테스트 케이스 개발에 활용하기 위한 테스트 프레임워크의 설계와 구현에 대해 소개하였다. 테스트 케이스 개발 난이도를 줄이고 개발 효율성을 증대시키기 위해 직관적이고 사용하기 쉬운 API 를 설계하였고, 테스트 결과를 일관되게 표시하기 위해 프레임워크에서 기본 UI 를 제공한다. 또한, 테스트 결과를 외부 서버에 전송하기 위해 테스트 리포터를 설계하고 구현하였다.

앞으로 TTA 에서는 스마트 TV 적합성 시험환경의 자동화와 멀티스크린과 같은 고급 기능 테스트를 위해 테스트 프레임워크의 고도화 개발을 추진할 계획이며, 이를 활용하여 스마트 TV 플랫폼 고급기능에 대한 테스트 케이스를 개발할 계획이다.

참고 문헌

[1] TTAK.KO.07-0111, “HTML5 기반 스마트 TV 플랫폼”, 2013.03

[2] Testharness.js Documentation Site, Available at, <http://testthewebforward.org/docs/testharness-documentation.html>

[3] TTAK.KO.07-0119, “ HTML5 기반 스마트 TV 플랫폼 수신기 표준 적합성 시험 ”, 2013.12

[4] Jasmine Documentation Site, Available at, <http://jasmine.github.io>

[5] jQuery user interface Site, Available at, <http://jqueryui.com/>