

DVB-MHP 환경에서의 내장형 DVB-HTML 브라우저

김 성기⁰, 홍 건호, 송 하윤, 한 상영

서울 대학교 컴퓨터 공학부, 홍익 대학교 정보 컴퓨터 공학부
{ditoman⁰, syhan}@ppplab.snu.ac.kr, {ghhong, song}@cs.hongik.ac.kr

An embedded DVB-HTML browser in DVB-MHP environment

SeongKi Kim⁰, Gunho Hong, Hayoon song, Sangyong Han
Seoul National University, Hongik University, Korea

요 약

데이터 방송을 셋톱 박스에서 시청하기 위해서 여러 가지 표준적인 플랫폼들이 존재하는데, 각 플랫폼마다 콘텐츠 표현 언어가 존재한다. 이러한 각 콘텐츠 표현 언어를 처리해주기 위해서는 데이터 방송용 브라우저가 필수적인데, 본 논문에서는 DVB-MHP 플랫폼에서 DVB-HTML 어플리케이션의 처리가 가능한 내장형 브라우저를 설계하고 구현 하였다.

1. 서 론

데이터 방송(Data Broadcasting)이란, 방송망을 통하여 방송 프로그램 관련 정보, 기상, 뉴스, 교통 등의 생활 정보를 전송하며, 인터넷, 전자 상거래까지 가능하게 해주는 기술을 말한다. 데이터 방송은 양 방향의 서비스를 제공한다는 점에서 기존의 방송과 구분되며, 방송망을 통해서 전송된다는 점과 방송과 함께 프로그램과 관련된 부가 정보까지 보내준다는 점에서 기존의 인터넷 방송과도 다르다.

데이터 방송을 시청하기 위해서는 셋톱 박스(Settop-Box)가 필요한데, 서로 다른 많은 셋톱 박스마다 표준적인 플랫폼들이 존재하여 어플리케이션과 콘텐츠의 개발, 표현을 쉽게 만들어준다. 이러한 플랫폼에는 ATSC-DASE (Advanced Television Systems Committee-DTV Application Software Environment), ARIB-STD B24 (Association of Radio Industries and Business-STD B24), OCAP (OpenCable Application Platform), DVB-MHP (Digital Video Broadcasting - Multimedia Home Platform)[1] 등이 존재한다.

각 플랫폼들마다 콘텐츠를 전송 받은 후, 화면에 옮

바르게 표시하기 위해서는 콘텐츠를 화면에 표시해주는 브라우저가 필수적인데, 본 논문에서는 DVB-MHP 환경에서 내장형 DVB-HTML 브라우저를 구현 하였다.

본 논문은 4장으로 이루어져 있는데, 2장에서는 DVB-MHP 플랫폼에 대해 기술하며, 3장에서는 DVB-MHP 환경에서 DVB-HTML 브라우저의 구현에 대해서 설명하며, 마지막 4장에서 결론을 내린다.

2. 관련 연구

2.1. DVB-MHP

DVB-MHP는 셋톱 박스에 많은 인터페이스들을 제공하며, 개발자는 이 인터페이스를 통하여 어플리케이션을 개발한다. DVB-MHP는 그림 1과 같은 구조를 가지고 있다.

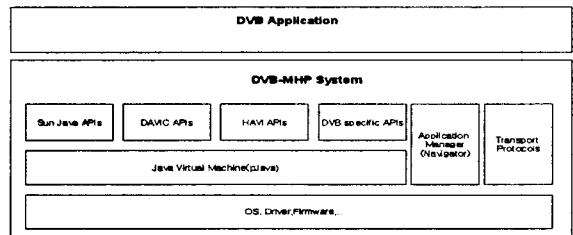


그림 1 DVB-MHP의 구조

그림 1과 같이 DVB-MHP는 OS위에 Java VM과 Sun Java API, DAVIC (Digital Audio Video Council), HAVI (Home Audio Video Interoperability)등의 각종 API들과 어플리케이션의 수명을 관리하는 어플리케이션 매니저, 통신 프로토콜 들로 구성되어 있다.

DVB-MHP에서 동작하는 어플리케이션에는 사용자 에이전트 의해 표현되는 DVB-HTML 어플리케이션과 Java VM에 의해 동작하는 DVB-J 어플리케이션이 존재한다. DVB-HTML 어플리케이션이 동작하는 어플리케이션 모델을 DVB-HTML 모델이라 하고, DVB-J 어플리케이션이 동작하는 어플리케이션 모델을 DVB-J 모델이라고 한다.

2.1.1. DVB-HTML 모델

DVB-HTML 어플리케이션은 리소스나 문서들의 모임이다. 여기에서 허용되는 리소스에는 DVB-HTML 콘텐츠, CSS 콘텐츠, ECMAScript 콘텐츠, 이미지, 동영상, 오디오 들이 포함된다.

DVB-HTML 어플리케이션은 사용자 에이전트에 의해서 수행 되어진다. 사용자 에이전트는 DVB-J 모델을 따르는 일반 어플리케이션이며, 여러 개의 액터를 가질 수 있다. 액터는 DVB-HTML 어플리케이션을 실제로 수행하는 곳 인데 Loading, Active, Paused, Killed, Destroyed 상태에 있으며, 수행과 관련된 여러 정보를 가지게 된다.

DVB-HTML 어플리케이션은 액터 안에서 Decoding, Presentation, Interaction 등의 과정을 거쳐 화면에서 처리된다. 어플리케이션의 상태는 액터의 상태와 동일한 상태를 가진다.

2.1.2. DVB-J 모델

DVB-J 어플리케이션은 Java VM에서 수행되며, 위에서 언급한 각종 API들을 사용해서 만들어지는 DVB-J 클래스들의 집합을 말한다. 어플리케이션 매니저의 제어 하에 시작하고 멈추며, Java TV의 Xlet 형식을 따라 만들어진다. Xlet 어플리케이션은 Loaded, Paused, Active, Destroyed 상태를 가진다.

3. 구현

3.1. 디자인

그림 2는 전체적인 브라우저의 구조이다.

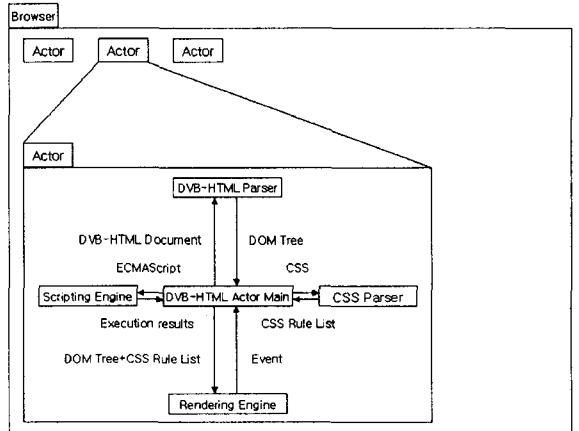


그림 2 DVB-HTML 브라우저의 구조

그림 2 에서 보는 것처럼 브라우저의 액터마다 DVB-HTML을 파싱하고 DOM 트리를 생성하는 DVB-HTML 파서, ECMAScript를 실행하는 스크립팅 엔진, CSS를 파싱 한 후 CSSStyleSheet를 생성하는 CSS 파서, DOM 트리를 받아서 화면에 그려주는 역할을 하는 렌더링 엔진, 다른 모든 부분 들을 제어해주는 역할을 하는 액터메인으로 구성된다.

3.1.1. DVB-HTML 처리

DVB-HTML 브라우저는 DVB-HTML 파일을 받으면 그림 3 처럼 DVB-HTML DOM 트리를 생성한다.

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//DVB/OTD XHTML DVB HTML 1.0/EN"
"http://www.dvb.org/mhp/dtd/dvbhtml-1-0.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:dvbhtml="http://www.dvb.org/mhp">
<head>
<title>This is Title</title>
</head>
<body>
This is sample body
</body>
</html>
```

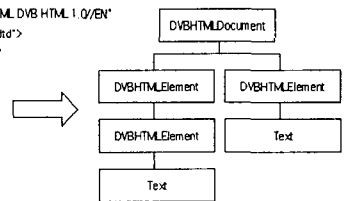


그림 3 DVB-HTML DOM 트리의 생성

DOM 트리를 생성하면서, style 엘리먼트를 만나거나 외부로 LINK 엘리먼트에 의해 링크된 CSS를 만나면, CSS 텍스트를 CSS 파서에게 파싱을 요청하여 스타일 시트를 검색할 수 있는 인터페이스를 제공하는 CSSStyleSheet를 생성한다.

또한 DOM 트리를 생성하면서 이벤트를 만나는 경우, EventListener를 생성하고 EventTarget의 addEventListener 메소드를 사용하여 등록된다.[2]

3.1.2. CSS 처리

하나의 엘리먼트에 적용 가능한 프로퍼티는 여러 CSS 소스에서 나타날 수가 있기 때문에, DVB-HTML 파서는 파싱하면서 엘리먼트의 적용 가능한 CSS 프로퍼티를 찾는다. DVB-HTML 안의 style 어트리뷰트, style 엘리먼트, 외부로 링크된 CSS 파일 안에서 순서대로 적용 가능한 프로퍼티를 찾고, 적용 가능한 프로퍼티가 존재하지 않는 경우, DVB-HTML의 기본적인 스타일 프로퍼티를 적용한다.

적용하는 방법은 엘리먼트는 DVBCSSInlineStyle을 구현하기 때문에, 해당 프로퍼티를 DVBCSSInlineStyle 인터페이스 안의 CSS2Properties 타입의 style 프로퍼티로 설정한다.

3.1.3. 렌더링

렌더링 엔진은 위의 과정을 거쳐 CSS가 연결된 DOM 트리를 전위 순회(In-order Traverse)하면서 추가된 CSS 프로퍼티에 따라 화면에 표시한다.

3.1.4. 이벤트 처리

화면에 DVB-HTML의 콘텐츠를 표시했으면 스크립팅 엔진이 스크립트를 실행한다. 특정 엘리먼트에 이벤트가 일어나는 경우, 브라우저는 Event를 생성한 후 엘리먼트에 등록된 EventListener들의 handleEvent 안으로 넘겨준다. handleEvent는 실제 ECMAScript를 호출하고 실행 결과를 가져온다.

3.2. 구현

위에서 디자인 한 DVB-HTML 브라우저를 구현하기 위해서 본 논문에서는 MHP 환경과 동일한 환경을 만들기 위해 Java VM으로 pJava 시뮬레이터 3.1, CSS 파서로 Steady state의 CSS2Parser 1.03 beta, ECMAScript 엔진으로써 Fesi 1.15, Sun의 Java TV 1.0, DVB-HTML 파서로써 Sun의 Jaxp 1.01을 사용 하

여 구현하였다.

구현 시에 브라우저는 DVB-J 모델을 따라야 하기 때문에 Loaded, Paused, Active, Destroyed의 상태를 가지지만 브라우저 안에 여러 개 존재하는 각각의 액터들은 Loading, Active, Paused, Killed, Destroyed의 상태에 존재하며, 각각 Decoding, Presentation, Interaction의 과정에 따라 처리한다.

4. 결론

데이터 방송을 시청하기 위해서는 셋톱 박스가 필수적인데 셋톱 박스에는 여러 가지 표준적인 플랫폼들이 존재한다. 이 플랫폼들 중에 DVB-MHP 플랫폼의 콘텐츠 표현 언어인 DVB-HTML 어플리케이션을 화면에 표시하는 브라우저를 일반 어플리케이션 모델인 DVB-J 모델에 맞춰서 본 논문에서는 디자인하고 구현 해 보았다.

여기서 구현한 브라우저는 DVB-MHP와 W3C의 DOM 인터페이스를 대부분 사용하며, Java로 구현되었다. Java로 구현되었기 때문에 비슷한 Java 환경인 ATSC-DASE로의 포팅이 용이하며, pJava를 사용하기 때문에 상당한 저 사양의 시스템 하에서도 동작이 가능하여 셋톱 박스와 같은 내장형 시스템에 적당하다.

5. 참고 문헌

- [1] ETSI, ETSI TS 102 812 Version V1.1.1 (2001-11) Digital Video Broadcasting; Multimedia Home Platform (MHP) Specification 1.1, November 2001
- [2] Tom Pixley, Netscape Communications Corp, Document Object Model (DOM) Level 2 Events specification Version 1.0, <http://www.w3.org/TR/DOM-Level-2-Events>, November 2000
- [3] Chris Wilson and Philippe Le Hégarret and Vidur Apparao, Document Object Model (DOM) Level 2 Style specification Version 1.0, <http://www.w3.org/TR/DOM-Level-2-Style/>, November 2000