

# Digital TV와 IPTV 환경을 위한 단말기 미들웨어 플랫폼 기술의 통합에 대한 연구

\*홍건호 \*\*송하윤

홍익대학교 컴퓨터공학과

\*gunho.hong@gmail.com

## A Study on the Integration of Embedded Middleware Platforms for IP and Digital TV Environments

\*Hong, Gun Ho \*\*Song, Ha Yoon

Department of Computer Engineering, Hongik University

### 요약

Digital TV와 IPTV 환경에서의 단말 어플리케이션은 디지털 콘텐츠에 대한 사용자 인터페이스를 제공하고 고부가가치의 복합 방송 서비스의 구현을 가능하게 하는 핵심 요소이며 이러한 단말 어플리케이션의 호환성을 확보하기 위해 방송 단말용 미들웨어 기술이 도입되었다. 방송 기술을 위한 단말 미들웨어는 크게 XHTML 브라우저 기반 플랫폼과 가상 머신 기반 플랫폼으로 양분되며 각각 독자적인 어플리케이션 모델을 제공한다. 이러한 두 가지 플랫폼의 통합을 위해 방송용 단말을 위한 미들웨어 국제 표준 기술 규격에서는 초기부터 브라우저와 가상 머신 플랫폼의 상호 운영성에 대해 정의하였다.

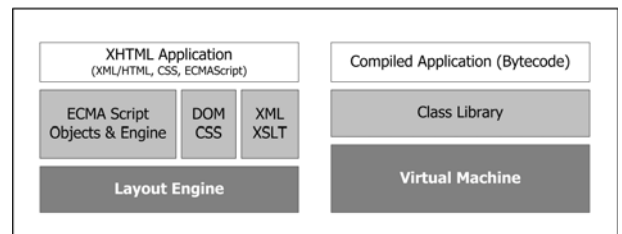
본 논문에서는 현재까지 다양한 플랫폼에서 연구 개발이 진행된 브라우저와 가상 머신 기반 미들웨어의 특성을 살펴보고 통합된 어플리케이션 모델과 이를 지원하는 미들웨어 아키텍처를 미들웨어 구현 관점에서 제시한다. 또한 국제 표준 규격들에서 제시하는 기술 방식을 검토하여 최근 변화한 기술 요소에 대한 적용점을 살펴본다. 마지막으로 TV 환경에서의 통합된 미들웨어 플랫폼 기술의 발전 전망과 방송용 단말 미들웨어 플랫폼에 대한 향후 연구에 관련된 주요한 기술적 변화들을 정리한다.

### 1. 서론

디지털 방송 인프라를 바탕으로 다양한 고부가가치의 방송 서비스를 제공하기 위해 사용자 단말에서 어플리케이션을 실행 가능 하도록 하는 미들웨어 기술이 도입되었다. 디지털 방송에서의 사용자 단말 어플리케이션은 디지털 콘텐츠에 대한 사용자 인터페이스를 제공하고 양방향 기능을 통해 고부가가치의 복합 방송 서비스의 구현을 가능하게 하는 핵심 요소이다.

방송용 단말 미들웨어는 크게 XHTML 브라우저 기반 플랫폼[1]과 가상 머신 기반 플랫폼으로 양분되며 각각 독자적인 어플리케이션 모델을 제공한다. 이러한 두 가지 플랫폼의 통합을 위해 방송용 단말을 위한 미들웨어 국제 표준 기술 규격에서는 초기부터 브라우저와 가상 머신 플랫폼의 상호 운영성에 대해 정의하였다. 그러나 미들웨어 탑재 단말기의 저조한 보급률, 고도화된 어플리케이션 서비스의 부재, 그리고 사용자 경험의 미비 등으로 인해 통합된 미들웨어 플랫폼 기술의 적용과 응용은 제한되어 왔고 이에 대한 기술적 논의와 규격화도 기초적인 상태에 머물러 왔다. 하지만 IPTV와 같은 컨버전스 환경의 등장으로 인해 digital TV 서비스가 경쟁적으로 확산되고, 단말기의 성능 및 기능 향상과 가격 하락 등이 진행되면서 방송용 단말 미들웨어에 대한 개발과 보급이 활성화 되고 단말 어플리케이션에 대한 상용화가 본격화 되고 있다. 이에 따라 상이한 어플리케이션 모델을 제공하는 브라우저와 가상 머신 기반의 미들웨어의 기술적 통합과 상호 운영 방안에 대한 기술적 논의가 필요하게 되었다[2].

### 2. 브라우저와 가상 머신 기반 플랫폼의 특성



<그림 1. 브라우저와 가상 머신 기반 플랫폼 구조>

디지털 TV와 IPTV 환경에서의 미들웨어 플랫폼은 크게 XHTML 기반의 브라우저 플랫폼과 Java 가상 머신 (Virtual Machine, 이하 VM) 기반 플랫폼으로 양분되어 발전 되어왔다[3]. 이 두 가지 플랫폼은 아래의 그림과 같이 상이한 어플리케이션 모델을 제공한다.

<표 1. 미들웨어 플랫폼 비교>

	XHTML 브라우저 플랫폼	Java VM 플랫폼
Programming 언어	ECMAScript, XHTML, CSS	Java
API	Native ECMAScript Objects, Document Object Model, XMLHttpRequest [4]	Java Profile API (Foundation Profile, Personal Basis Profile, Java TV, etc)
실행방식	Interpreted / JIT	Compiled / JIT

\* 이 논문은 2008학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음

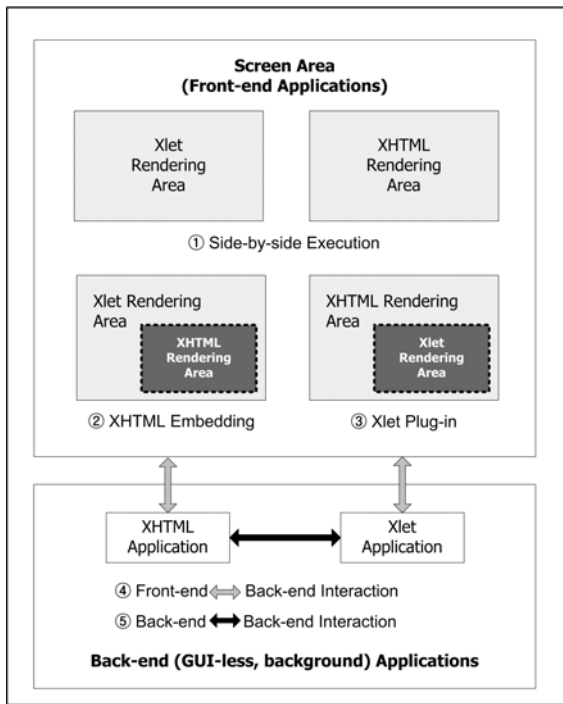
어플리케이션 구성단위	XHTML Page	Xlet Package
관련 표준	EMCA-262 [5] W3C XHTML, DOM, CSS specification	Java Language Specification [6], Java Virtual Machine specification [7]

### 3. 브라우저와 가상 머신 기반 플랫폼의 통합 방안

XHTML 브라우저 기반의 미들웨어와 Java VM 기반 미들웨어 플랫폼의 통합 아키텍처를 디자인하기 위해서는 먼저 통합된 미들웨어에서 지원되어야 하는 어플리케이션 모델들을 정의하고 이를 기반으로 아키텍처가 고려되어야 한다.

#### 가. 통합 플랫폼이 지원하는 어플리케이션 모델

어플리케이션의 응용 특성에 따라 다음과 같은 5 가지의 통합된 어플리케이션 모델을 정의할 수 있다.



<그림 2. 통합 미들웨어의 어플리케이션 모델>

##### ① 독립된 실행 모델

XHTML 어플리케이션과 Xlet 어플리케이션이 각각 독립적인 렌더링 영역을 가지고 실행하는 모델

##### ② XHTML 내장(Embedding) 모델

Xlet 어플리케이션의 렌더링 영역 안에 XHTML 어플리케이션의 콘텐츠를 표시하는 모델

##### ③ Xlet 플러그인(Plug-in) 모델

XHTML 어플리케이션이 Xlet 어플리케이션을 플러그인 형태로 동작시켜 특정 영역에 대한 렌더링을 수행하는 모델

##### ④ Front-end 와 Back-end 상호 작용 모델

렌더링 영역을 가지고 Front-end 기능을 수행하는 XHTML 또는 Xlet 어플리케이션이 Back-end에서 독립적으로 실행되는 XHTML 또는 Xlet과 연동하는 모델. XHTML 어플리케이션의

경우 ECMAScript가 독자적으로 실행되거나 숨겨진 XHTML 페이지가 지속적으로 브라우저에 의해 실행되는 구조이다.

##### ⑤ Back-end 어플리케이션 간 상호 작용 모델

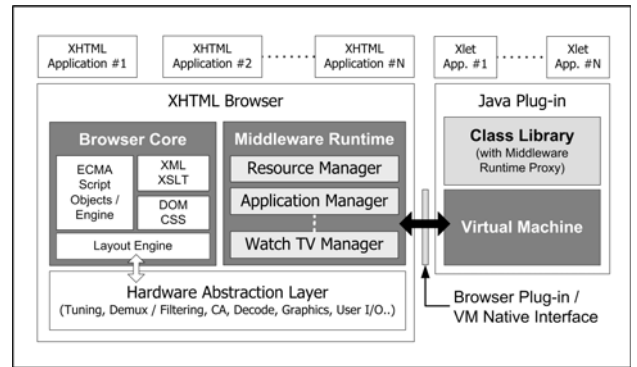
Back-end에서 독립적으로 실행되는 Xlet 어플리케이션 또는 XHTML 어플리케이션 사이에 정의되는 연동 모델

#### 나. 미들웨어 구현 관점에서의 통합 아키텍처

앞서 논의된 어플리케이션 모델들을 지원하기 위한 미들웨어 아키텍처는 플랫폼 상에서 실행되는 다수의 어플리케이션 life-cycle과 자원을 관리하는 어플리케이션 매니저 등이 포함되는 미들웨어 런타임의 구현 주체에 따라 브라우저 기반의 통합, 가상 머신 기반의 통합 그리고 독립된 실행 환경 기반의 통합 구조로 정의될 수 있다.

##### ① 브라우저 기반의 통합구조

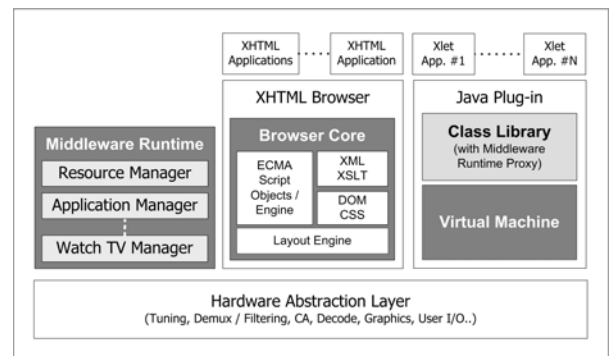
브라우저는 미들웨어 런타임을 통해 하드웨어의 제어 및 자원 관리를 수행하며 XHTML 어플리케이션 상의 요청(태그 또는 ECMAScript를 통한 동적인 객체 생성)이나 외부 Signaling (데이터 방송 신호의 수신 등, 예를 들어 AIT)에 의해 브라우저에 연계된 가상 머신을 구동한다.



<그림 3. 브라우저 기반 미들웨어 통합 구조>

##### ② 가상머신 기반의 통합 구조

브라우저를 Java 기반 플랫폼에서 내장(Embedding)하거나 가상 머신 상에 브라우저를 직접 구현하는 형태이다.



<그림 4. 독립된 런타임 기반 미들웨어 통합 구조>

##### ③ 독립된 런타임 기반 통합 구조

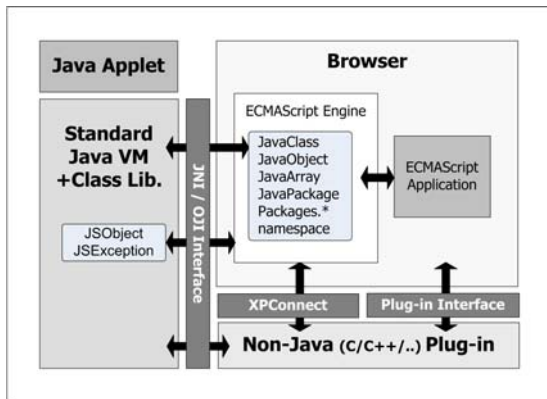
브라우저와 가상 머신 기반 미들웨어가 독립적으로 실행되면서 별도의 미들웨어 런타임의 관리하에 상호 필요에 따라 Plug-in 또는 Embedding 모드로서 동작하는 형태이다.

#### 4. 미들웨어 표준 규격의 한계와 개선 방안

MHP, ACAP, 그리고 OCAP 등의 표준 미들웨어 규격들은 초기 단계부터 Java 기반 어플리케이션 모델과 XHTML 기반 어플리케이션 모델을 함께 고려하여 제정되었다. 그러나 실질적으로 표준 규격의 개발에 있어서 Java 기반 API를 정의하고 검증하는 것이 중심이 되어왔기 때문에 상대적으로 XHTML 관련 사항에 대한 규격화는 초기 정의에서 큰 변화 없이 진행되어 왔다.

현재까지 가장 성공적으로 정의되고 개발되어온 MHP (Multimedia Home Platform) [8] 미들웨어 규격에서는 XHTML 어플리케이션 (DVB-HTML)의 ECMA Script와 Java Application (DVB-J)의 연동을 위해 LiveConnect [9] 기술을 기반으로 규격을 정의하고 있다.

LiveConnect 규격은 Java 플러그인과 ECMAScript 간의 연동을 위해 Netscape사에 의해 개발되었으며 Navigator 브라우저와 이후에 개발된 Mozilla 프로젝트 기반 브라우저에 그 구조가 유지되어 왔다. 다음 그림은 Mozilla 기반 브라우저에서 LiveConnect를 비롯한 연동 구조를 보여준다.



<그림 5. LiveConnect 기반의 브라우저 구조>

이러한 현재 미들웨어 규격의 한계는 LiveConnect 기술의 구현 특성과 기술 규격의 변화에 있다. LiveConnect를 구현한 Mozilla 프로젝트에서도 지속적으로 구현 방법의 변경을 거듭해 오다가 최근 LiveConnect 관련 형상을 향후 버전에서 변경하기로 결정하였으며 [10] Sun microsystem의 Java VM 구현 방식도 새로운 Plug-in 형태의 구현을 통해 브라우저 간 표준화된 Plug-in 모델 기반으로 LiveConnect를 구현하는 것으로 변경 되었으며 이에 관련된 새로운 규격을 발표하였다[11]. 따라서 Mozilla 프로젝트에서 유지되던 LiveConnect 규격을 채용한 미들웨어 규격도 현재 기술 변화를 반영하는 규격 개정이 요구된다.

#### 5. 결론과 향후 관련 연구

본 논문에서는 디지털 TV와 IPTV 환경에서 다양한 어플리케이션 모델을 지원하기 위한 XHTML 브라우저 기반 미들웨어와 Java VM 기반 미들웨어의 기술적인 통합 방안을 고려해 보고 현재 표준 기술 규격의 문제점을 살펴보았다.

방송용 단말 미들웨어 플랫폼에 대한 향후 연구에 관련된 주요한 기술적 변화를 살펴본다면, 먼저 차기 버전의 ECMA Script의 표준화 작업[12]과 더불어 가상 머신 기반으로 구현된 ECMA Script 엔진에 대한 성능 개선에 대한 연구 결과들이 도출되고 있

고 이를 통해 브라우저 플랫폼에서는 기존에 화면 렌더링을 담당하는 Layout engine보다 ECMA Script engine이 Web 2.0 트렌드와 맞물려 중심적인 요소가 되어가고 있다는 것이다.

Java 기반 미들웨어 플랫폼 관점에서는 어플리케이션의 효과적인 자원관리를 위한 Multi Java VM instance 기반의 어플리케이션 격리(isolation) 기법과 Java VM 상에서의 스크립트 실행 기능이 향후 미들웨어 규격의 변화에 영향을 미칠 것으로 예상된다.

또한 기존의 XHTML과 Java 기반 어플리케이션 모델 뿐 아니라 Flash 등의 신규 플랫폼이 방송 분야에 적용됨에 따라 이에 대한 효과적인 연동과 통합 구조에 대한 연구가 필요할 것이다.

마지막으로 현재 진행 중인 다양한 모바일 플랫폼간의 치열한 경쟁과 확산에 있어서 각 플랫폼의 발전된 기능들이 방송용 단말의 미들웨어 기술에도 영향을 줄 것으로 예상되며 이러한 모바일 플랫폼과 연관된 연구 개발의 진행을 계획하고 있다.

#### 6. 참고 문헌

- [1] Hun Lee, Gun Ho Hong, Ha Yoon Song and Sang Yong Han, Coding and Presentation of Multimedia for Data Broadcasting with Broadcasting Markup Language, Lecture Notes in Computer Science Volume 2544, 2002 Databases in Networked Information Systems, 147-160
- [2] Ha Yoon Song; Jongwook Park, Design of an interoperable middleware architecture for digital data broadcasting, Consumer Electronics, IEEE Transactions on Volume 52, Issue 4, Nov. 2006 Page(s):1433 - 1441
- [3] 김성기,홍건호,송하운,한상영, "DVB - MHP 환경에서의 내장형 DVB - HTML 브라우저", 한국정보과학회 2003년도 봄 학술발표논문집 제30권 제1호, 2003년 4월, pp. 518~520
- [4] The XMLHttpRequest Object, W3C Working Draft 15 April 2008 (<http://www.w3.org/TR/2008/WD-XMLHttpRequest-20080415/>)
- [5] ECMAScript Language Specification (ECMA-262), 3rd Edition, Ecma international, December 1999
- [6] The Java Language Specification, 3rd Ed., Sun microsystems, 2005 (<http://java.sun.com/docs/books/jls/>)
- [7] The Java Virtual Machine Specification, 2nd Ed., Sun microsystems, 1999 (<http://java.sun.com/docs/books/jvms/>)
- [8] Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP) Specification 1.2 (DVB Document A107), DVB Project, 2007
- [9] LiveConnect on the mozilla developer center (<https://developer.mozilla.org/en/LiveConnect>)
- [10] Disable OJI (LiveConnect) in mozilla-central ([https://bugzilla.mozilla.org/show\\_bug.cgi?id=397080](https://bugzilla.mozilla.org/show_bug.cgi?id=397080))
- [11] LiveConnect Support in the New Java™ Plug-In Technology (<https://jdk6.dev.java.net/plugin2/liveconnect/>)
- [12] Proposed ECMAScript 4th Edition (Language Overview), Revised 23 October 2007 (<http://www.ecmascript.org/es4/spec/overview.pdf>)