

개방형 클라우드 방송 기반 스마트 방송 애플리케이션의 형식과 관계를 위한 모델 정의¹

김호년, 임성묵, 김정현, 이동훈, 최미애, 박동영
한국정보통신기술협회

{ hykimfnd, seongmook.lim, jllh, dhlee, miae, dypark }@tta.or.kr

Model Definition for Type and Relation of Open Cloud-Based Smart Broadcasting Application

Ho-Youn Kim, Seong-Mook Lim, Jung-Hyun Kim, Dong-Hoon Lee, Miae-Choi, Dong-Young Park
Telecommunications Technology Association

요 약

본 논문은 “개방형 클라우드 기반 스마트 방송 플랫폼” 표준에 따라서 개발될 수 있는 다양한 클라우드 기반 스마트 방송 애플리케이션의 형식과 관계를 규정하기 위한 애플리케이션 모델을 소개한다. 스마트 방송용 애플리케이션이 다양한 클라우드 방송 플랫폼에서 일관된 사용자 경험을 제공하며 동작하기 위해 클라우드 방송 환경의 특성을 고려해야 하며, 이에 따라 애플리케이션의 형식을 분류하고 실행 및 종료 방법과 생명 주기를 정의하며 애플리케이션 사이에서 동작 관계를 규정하는 등의 모델 정립이 필요하다. 본 논문에서는 클라우드 방송 애플리케이션을 방송 수신 기능의 활성화 여부에 따라 방송 독립형 및 연동형으로 분류하고, 각 애플리케이션의 생성과 종료를 위한 생명주기와 각 애플리케이션 사이에서 우선 순위를 기반으로 한 상호 작용 정책을 규정한다. 또한, 클라우드 애플리케이션의 형식 및 우선순위 등을 애플리케이션에 지정할 수 있는 관리 정책을 정의한다. 이러한 모델 정의를 바탕으로 표준을 적용한 클라우드 방송 테스트 플랫폼을 구현하고 다양한 형태의 클라우드 방송 애플리케이션을 개발하여 모델 정의에 따르는 기본 동작 시나리오 및 유효성을 검증한 사례를 소개한다.

1. 서론

클라우드 컴퓨팅 및 네트워크 기술을 발달로 스토리지, 분산 서버, 사용자 애플리케이션 등 다양한 분야에서 클라우드 기술이 접목되고 있다. 클라우드 기술은 고속 네트워크 및 고성능 서버의 분산 연산 기술을 기반으로 서버에서 애플리케이션 및 운영체제 등의 소프트웨어를 실행하고 실행 결과를 사용자 단말에 전송하여 표시함으로써 최소한의 사용자 단말 성능으로 높은 연산량을 요구하는 프로그램을 실행할 수 있도록 한다.

최근 이러한 클라우드 기술은 방송의 스마트미디어 분야에도 적용되고 있다. 특히, 방송 수신기의 서비스 실행기능을 가상화하여 클라우드 서버에서 실행하고 실행 결과 화면을 비디오 혹은 이미지로 수신기에 전송함으로써 저사양의 수신기에서도 고사양의 실행 성능을 요구하는 스마트미디어 서비스를 실행할 수 있다. 이러한 기술은 국내 유료방송 업계에서 상용화 되어 점차 적용범위가 확대되고 있다.

한편, 국내 클라우드 기반 방송 기술의 확산과 스마트미디어 서비스 활성화를 지원하기 위해 한국정보통신기술협회(TTA)

에서는 웹을 기반으로 클라우드 서버의 서비스 실행영역을 정의하고 클라우드 서버와 수신기간 전송 기술을 표준화하는 “개방형 클라우드 기반 스마트 방송 플랫폼” 표준 [1] (이하, 클라우드 방송 표준)을 제정하였다.

본 논문은 클라우드 방송 표준에서 표준기반의 다양한 클라우드 환경의 수신기에서 소스코드 수정 없이 동일한 사용자 경험을 제공하기 위해 클라우드 방송 환경에서 실행되는 애플리케이션에 대한 형식과 실행 및 종료 시나리오를 규정하는 애플리케이션 모델에 대해서 소개한다.

본 논문은 다음과 같이 구성된다. 제 2 장에서는 표준의 구성과 W3C/HTML5 기반 실행환경 정의, 서비스 전송 기술 및 수신기 연동 프로토콜 등 표준에 대한 내용을 간략히 소개한다. 제 3 장에서는 클라우드 방송 환경에서 실행되는 애플리케이션의 분류에 대해 소개하고, 제 4 장에서는 애플리케이션 분류에 따른 실행 및 종료 시나리오에 대해 소개한다. 제 5 장에서는 애플리케이션의 분류, 자원, 권한 등에 대한 관리 정책에 대해 소개하고, 제 6 장에서는 클라우드 방송 표준을 구현한 플랫폼과 애플리케이션을 통해 검증한 사례를 소개한다. 마지막으로 제 7 장에서는 논문의 결론 및 향후 과제를 제안한다.

¹ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [2017-0-00986, 방송융합 기반기술 테스트 환경구축]

2. 클라우드 방송 표준 소개

클라우드 방송은 서버에서 방송 서비스를 구동한 후 실시간으로 실행 화면을 캡처하여 압축 이미지 혹은 압축 비디오 형태로 수신기에 스트리밍함으로써 이를 디코딩하여 렌더링하는 최소한의 수신기 성능으로 방송 서비스를 실현할 수 있는 수신기 가상화 기술이다. 그림 1은 기존 방송과 클라우드 방송 서비스의 실행 및 전송 과정을 비교하여 나타낸다.

그림 1 기존 방송과 클라우드 방송 서비스 비교



2016년 12월 TTA에서는 이러한 클라우드 방송 기술을 특정 기술에 종속되지 않고 표준기반으로 확산시키기 위한 목적으로 클라우드 서버 플랫폼의 실행 환경, 클라우드 방송 전송 기술 및 수신기와의 연동 인터페이스 등을 정의하는 클라우드 방송 표준을 제정하였다.

이 표준에서 클라우드 서버 플랫폼의 애플리케이션 실행 환경은 W3C/HTML5 웹 기술을 기반으로 한다. 스마트 TV 플랫폼의 웹 코어 기술을 정의한 OIPF의 'Web Standards TV Profile' [2] 및 "HTML5 기반 스마트 TV 플랫폼" [3] 표준을 준용하고, 최근 W3C에서 표준개발이 완료되어 주요 브라우저 개발사에서 적용하고 있는 HTML5 및 CSS3의 기술을 추가적인 요구사항으로 정의하고 있다. 특히, 웹에서 게임과 가상 콘텐츠와 같은 고성능의 그래픽 효과가 필요한 서비스를 개발할 수 있도록 Web GL [4]을 지원하도록 규정하고 있다. 또한, W3C 웹 기술에서 정의하지 않는 TV 제어 기능을 지원하기 위해 "HTML5 기반 스마트 TV 플랫폼" 표준의 확장 인터페이스를 준용하고 클라우드 방송 실행 환경을 고려하여 일부 인터페이스 기능은 제외하고 방송 영상 제어 API 및 고객 정보 조회 API 등을 추가로 정의하고 있다.

이와 더불어, 클라우드 방송 표준에서는 클라우드 서버와 수신기 간 클라우드 서비스의 전송 방식을 표준화 하며, 전송 데이터의 종류에 따라 이미지 전송 방식, 비디오 전송 방식, 하이브리드 전송 방식(이미지와 비디오 전송 방식의 융합된 형태)으로 구분하여 전송포맷과 전송 데이터를 정의하고 있다. 또한, 수신기에서의 입력 장치의 입력 데이터의 전송에 대한 전송 방식도 함께 정의한다. 그리고, 클라우드 서비스에서 확장 API를 호출하거나 수신기의 초기 접속 과정에서 전송되는 정보 등 클라우드 서버와 수신기 간에 연동을 위해 필요한 메시지 교환 형식과 전송 메시지를 정의하기 위해 수신기 연동 프로토콜을 기술하고 있다.

3. 애플리케이션 형식

클라우드 방송 애플리케이션은 애플리케이션 실행 시 수신기의 기본 동작 상태에 따라 방송 연동형 애플리케이션과 방송 독립형 애플리케이션으로 분류한다. 표 1은 클라우드 방송 애플리케이션 형식의 요약과 실행 예를 설명한다.

방송 연동형 애플리케이션은 클라우드 방송 서비스를 위한 애플리케이션이 실행이 될 때, 방송 비디오와 오디오 같은 수신기의 방송 자원의 상태가 그대로 유지되면서 실행되는 애플리케이션의 형식이다. 이 형식의 애플리케이션이 실행이 될 때 방송수신기는 현재 실시간 방송의 상태를 중지하거나 종료해서는 안되며, 애플리케이션은 실시간 방송에 영향을 주지 않으면서 방송과 함께 실행되어야 한다. 이 형식의 애플리케이션 서비스는 실시간 방송 비디오 위에서 오버레이(Overlay)된 상태로 실행될 수 있고, 애플리케이션이 API 호출에 의해서 방송 비디오의 크기와 위치를 변경하여 PIP(Picture In Picture) 형식으로 실행될 수도 있다.

반면, 방송 독립형 애플리케이션은 클라우드 방송 서비스를 위한 애플리케이션이 실행이 될 때, 방송 비디오와 오디오 같은 수신기의 방송 자원의 상태를 종료하거나 중지한 상태로 실행되는 애플리케이션의 형식이다. 이 형식의 애플리케이션이 실행이 되면, 수신기는 현재 재생하고 있는 실시간 방송서비스를 멈추고 기본 화면 상태(ex.검은색 백그라운드 배경화면)에서 앱을 실행하여야 한다. 클라우드 방송 환경에서 VOD나 게임과 같이 현재 실행하고 있는 실시간 방송과는 무관하게 클라우드 서비스를 제공하는 애플리케이션의 경우에는 방송 독립형으로 실행이 되어야 하며, 독립형 애플리케이션이 실행하는 동안 정지하고 있던 방송 서비스는 애플리케이션이 종료하고 나면 다시 원상태로 복원이 되어야 한다.

표 1 클라우드 방송 애플리케이션 형식 정의 요약

구분	방송 연동형 애플리케이션	방송 독립형 애플리케이션
기본 동작	클라우드 방송서비스 제공자 AIT를 통한 앱 실행	클라우드 서비스 메뉴(앱)를 통한 앱 실행
명시적 지정	애플리케이션 관리 정책에 앱 형식이 지정되어 있으면, 지정된 애플리케이션의 형식을 준수함 (※ 명시적 지정방법이 기본동작보다 우선순위가 높다)	
실행 예		

4. 애플리케이션 생명주기와 우선순위

클라우드 방송에서 애플리케이션은 다양한 방법으로 시작되고 종료될 수 있으며, 하나의 애플리케이션이 실행되고 종료되는 과정을 생명주기라고 한다. 클라우드 방송 플랫폼에서 애플리케이션의 안정적인 실행과 다양한 클라우드 방송 환경에서 동일한 사용자 경험을 제공하기 위해 애플리케이션 생명주기는 명확히 정의되어야 한다.

애플리케이션이 실행과 종료는 애플리케이션의 종류, 관리 정책, 애플리케이션의 구현 등에 따라 결정되며, 본 논문에서는 표 2와 같이 애플리케이션의 실행 및 종료되는 경우를 정의한다.

이러한 애플리케이션 생명주기를 바탕으로 동시에 여러 애플리케이션이 실행될 수 있는 조건에서 우선 실행되어야 하는 규칙을 정의할 필요가 있으며, 본 논문에서는 아래와 같이 정의한다.

- 사용자의 메뉴 선택에 의한 애플리케이션의 실행은 애플리케이션 시그널링에 의한 애플리케이션 실행보다 우선순위가 높다.
- 애플리케이션 관리 정책에 우선순위가 지정되어 있는 경우 이를 따른다.
- 시그널링되는 앱 간의 우선순위는 시그널링 정보의 우선순위를 따른다.

표 2 애플리케이션 실행/종료되는 경우

앱이 실행되는 경우	앱이 종료되는 경우
<ul style="list-style-type: none"> • 애플리케이션 시그널링에 의한 앱 실행 • 사용자의 클라우드 서비스 메뉴를 통해 선택하여 앱 실행 • 확장 API 호출을 통한 앱 실행 	<ul style="list-style-type: none"> • 앱 실행 중 확장 API 호출을 통해 앱 종료 • 시그널링 정보에 앱 종료 명령이 있는 경우 앱 종료 • 사용자가 리모콘의 종료(EXIT) 키를 인가하는 경우 실행 중인 앱 종료 • 클라우드 서비스 예외 상황 발생 시 앱 종료

5. 애플리케이션 관리 정책

본 논문에서 정의하는 애플리케이션 관리 정책은 클라우드 방송 환경에서 서비스를 제공하는 애플리케이션이 실행 및 종료할 때 요구되는 관리 항목을 정의하는 정책과 그 값을 정의하는 것으로, 클라우드 방송 서비스 제공자의 사업 정책 및 환경에 맞게 서비스를 제공하기 위한 수단으로 활용될 수 있다. 클라우드 방송 서비스 애플리케이션은 실행이 되면서 애플리케이션 관리 정책에 따라서 실행 문맥(Execution Context)을 생성하고 이 문맥을 통해 애플리케이션의 실행에 필요한 자원과 권한 등을 부여 받게 된다. 애플리케이션이 실행되면 플랫폼은 관리 정책이 기술된 파일을 조회하고, 해당 관리 정책에 따라서 애플리케이션의 실행에 필요한 환경 설정을 하게 된다.

이러한 애플리케이션 관리 정책은 클라우드 방송 표준 7.5 절에 기술된 XML 스키마가 정의하는 애플리케이션 관리 정책 테이블에 기술되며, 애플리케이션의 시작 페이지와 동일한 위치에 AMPT 로 파일명이 시작되는 xml 확장자의 파일로 저장된다.(예, AMPT_example.xml) 클라우드 서버 플랫폼은 이를 통해 애플리케이션에 필요한 관리 정책을 실행 환경에 반영한다. 표 3 은 애플리케이션 관리 정책에서 지정하는 정책의 종류를 설명한다.

표 3 애플리케이션 관리 정책 구성 항목

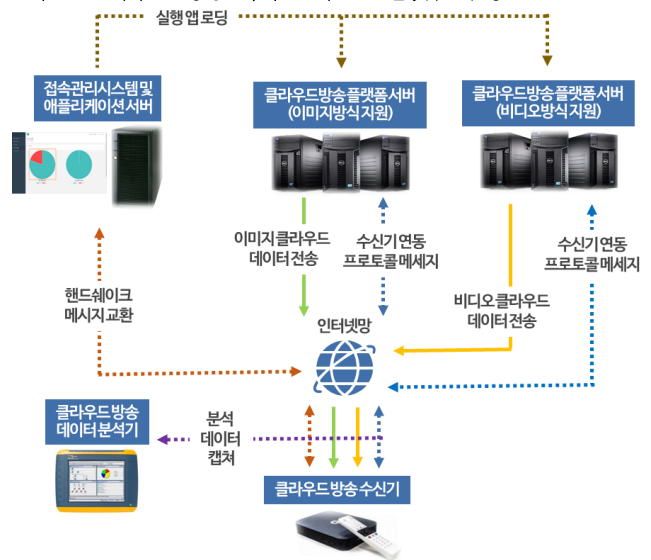
항목	설명
실행 형식	애플리케이션의 실행 형식을 <apptype> 속성을 통해 지정, 방송 연동형인 경우 'related' 값을 독립형인 경우 'independent' 값을 설정

우선순위	애플리케이션의 우선순위를 명시하기 위한 것으로 <apppriority> 요소의 값(양의 정수 값)으로 설정
리모컨 키 속성	애플리케이션이 전달받을 리모컨 키 정보를 <remotekey> 요소의 값으로 설정
수신기 자원 접근 권한	애플리케이션이 접근하고자 하는 수신기의 자원을 명시하기 위한 것으로 <permission> 속성값을 설정
위치 지정	AMPT 는 애플리케이션 시작 페이지와 동일한 위치에 AMPT 로 시작하는 XML 파일로 저장되며, 별도의 위치에 저장할 경우 <redirection> 요소에 그 위치값을 기술하여 설정함

6. 구현 및 검증

본 논문에서는 제안한 애플리케이션 모델의 타당성을 검증하기 위하여 표준 및 앞서 기술한 내용을 적용한 클라우드 방송 서비스 테스트 플랫폼을 구현하였다. 또한, 논문에서 제시한 모델의 다양한 애플리케이션을 개발하여 클라우드 방송 플랫폼 서버 및 수신기의 동작을 확인하였다. 그림 3 은 구현된 클라우드 방송 서비스 테스트 플랫폼의 구성을 나타낸다.

그림 2 클라우드 방송 서비스 테스트 플랫폼 구성



본 테스트 플랫폼은 클라우드 방송 표준을 구현한 클라우드 방송 플랫폼 서버와 수신기 간 접속을 관리하고 수신기 인증 등의 핸드셰이크(handshake) 과정을 지원하는 접속관리시스템, 클라우드 애플리케이션 및 AMPT 파일을 호스팅하고 관리하는 애플리케이션 서버, 클라우드 방송 플랫폼 서버에 접속하여 클라우드 애플리케이션을 수신할 수 있는 수신기, 그리고 테스트 플랫폼 구성요소 간 송수신 데이터를 분석하기 위한 클라우드 방송 데이터 분석기 등으로 구성되어 있다. 본 논문에서 제안한 애플리케이션 모델은 클라우드 서버의 앱 실행환경에 웹코어를 확장한 형태로 구현하였다.

그림 2 는 리모콘의 메뉴 키를 입력하여 클라우드 서비스 메뉴 화면으로 진입한 화면과 수신기를 나타내고 있다. 그림

3 은 클라우드 서비스 메뉴에서 비디오 전송 방식의 방송 독립형 애플리케이션을 실행한 모습을 나타낸다. 그림 4 는 애플리케이션이 실행되는 채널에 진입하여 방송 연동형 앱이 실행되는 모습을 나타낸다.

그림 3 클라우드 서비스 메뉴 및 클라우드 방송 수신기



그림 4 비디오 전송 방식의 방송 독립형 앱 실행 예



그림 5 이미지 전송 방식의 방송 연동형 앱 실행 예



7. 결론

본 논문에서는 개방형 클라우드 기반 스마트 방송 플랫폼의 애플리케이션 실행환경에서 클라우드 방송 애플리케이션이 플랫폼 구현에 상관없이 일관되게 동작할 수 있도록 정의한 애플리케이션 모델을 소개하였다. 클라우드 방송 환경을 고려하여 방송 독립형 및 연동형으로 애플리케이션을 분류하였고, 이를 바탕으로 애플리케이션의 생명주기와 앱 실행 우선순위 규칙을 정립하였다. 그리고, 이러한 모델과 클라우드 방송 표준을 구현한 테스트 플랫폼을 통해 논문에서 제안한 내용을 검증하였다.

향후 애플리케이션 모델은 수신 단말 종류의 다양화, 멀티 애플리케이션 실행, 수신기 네이티브 애플리케이션과의 연동 등을 고려하여 확대 정의될 필요가 있을 것이다.

앞으로 TTA 에서는 클라우드 방송 표준에 대한 적합성 시험 환경을 구축하고 테스트 도구 개발 및 시험 서비스를 제공하여 표준을 준수하는 다양한 클라우드 플랫폼과 앱 사이에서의 상호운용성 검증을 지원할 예정이다. 이와 더불어, 클라우드 방송 표준이 고급 기능을 지원할 수 있도록 멀티스크린, 결제인터페이스, 광고삽입 등에 대한 표준화를 추진하며 동시에 보안 및 전송 방식 확대 등 기존 내용을 보완할 수 있는 표준기술 개발도 추진할 계획이다.

참고 문헌

- [1] TTAK.KO.07-0129, “ 개방형 클라우드 기반 스마트 방송 플랫폼 ”, 2016.12
 - [2] OIPF Release 2 Specification, ‘ Volume 5a - Web Standards TV Profile [V2.3] ’, 2014.01
 - [3] TTAK.KO.07-0111/R2, “ HTML5 기반 스마트 TV 플랫폼 ”, 2016.06
 - [4] Khronos, WebGL, 2016. 5.
- <https://www.khronos.org/registry/webgl/specs/latest/1.0>