데이터방송 멀티 애플리케이션 개발

*김현순 **권재광 ***강대갑

한국방송 방송기술연구팀

*soon71@kbs.co.kr

Development of Multi-applications for Data Broadcasting

*Kim, Hyun-Soon **Kwon, Jae-Kwang ***Kang, Dae-Kap

*Broadcast Technical Research Team, Korean Broadcasting System(KBS)

요약

단일 애플리케이션 송출 환경에서의 데이터방송에서는 특정 시간, 특정 채널에 대해 하나의 애플리케이션만을 서비스할 수 있었다. 이러한 단점을 극복하기 위하여 2005년 KBS(Korean Broadcasting System)에서는 ACAP(Advanced Common Application Platform) 규격을 따르는 범용적인 멀티 애플리케이션 구조/운용 방법을 개발하고 이를 지원하도록 데이터방송 시스템을 개선하였다. 멀티 애플리케이션 서비스가 제공되면 방송사가 특정 시간, 특정 채널에서 하나 이상의 애플리케이션을 자유롭게 편성하여 서비스하므로 사용자는 다양한 애플리케이션을 기호에 맞게 선택, 이용할 수 있다.

본 논문에서는 추후 실제 본 방송에 적용하기 위하여 개발한 실험용 멀티 애플리케이션을 중심으로 살펴보고자 한다. 애플리케이션 측면에서 멀티 애플리케이션의 핵심은 일반 애플리케이션들에 대한 관리를 담당하는 매니저(혹은 루트) 애플리케이션이므로 이에 대한 기능 및 구조에 대하여 먼저 살펴 보고, 방송 송출 시스템 및 수신기에 대한 정합 실험을 위하여 개발된 멀티 애플리케이션 개발 현황에 대하여 살펴 본다.

1. 서론

방송의 디지털화로 TV를 보면서 경매, 게임, 물품 구매, 부가 정보 보기 등의 데이터방송 서비스를 사용할 수 있게 되었다. 지상파 데이터방송 표준인 ATSC(Advanced Television Systems Committee)-ACAP[1, 2]이 표준으로 확정됨에 따라, 지상파 방송사들은 실험 방송을 통하여 수신기 및 방송 송출 시스템에 대한 기술적인 정합 실험을 실시하고 다양한 애플리케이션을 서비스하면서 본 방송 대비에 박차를 가하고 있다[3].

그러나 대부분 특정 시간, 특정 채널에 대해 하나의 애플리케이션만을 서비스하고 있거나 방송사에서 하나 이상의 애플리케이션을 송출하고 있다고 하여도 특정 수신기에서는 그 중 하나만 수신되어 서비스되는 등으로 인하여 멀티 애플리케이션에 대한 필요성이 대두되고 있다.

즉 현재 지상파 방송사들의 데이터방송 실험 방송에는 애플리케이션 측면에서 멀티 애플리케이션이라는 개념이 도입되지 않아, 여러 개의 애플리케이션을 실험 서비스하는 방송사에서도 단순히 일반 단일 애플리케이션 여러 개를 동일 시간, 동일 채널에 송출하는 형태로 서비스하고 있다. 이 경우, 단일 애플리케이션 여러 개를 서비스하는 방송사는 사용자가 원하는 애플리케이션을 선택하여 시청할 것을 기대하지만, 수신기의 종류에 따라 그 서비스를 수행하지 않을 수도 있다. 국내 특정 수신기의 경우 여러 개의 애플리케이션이 송출되고 있을 때 이들의 이름을 보여주는 단순한 애플리케이션 리스트를 보여주고 사용자가 선택하여 실행할 수 있도록 하고 있는 경우도 있지만, 하나의 애플리케이션만을 수신하여 보여주는 수신기를 가지고 있는 사용자는 다수의 애플리케이션들이 서비스되고 있는지조차 알 수 없을 것이다.

만약 모든 수신기에서 애플리케이션 리스트 기능을 제공한다고 해도 수신기마다 리스트를 보여주는 방법, 구체적인 처리 방법 등이 달라 사용자에게 혼란을 초래할 것이다. 또한 수신기가 제공하는 애플리케이션 리스트 UI(User Interface)는 애플리케이션 이름들을 나열하여 선택하도록 하는 정도의 수준에서 크게 벗어나지 못한다. 그러므로 방송사는 자사의 특성을 반영하여 다양한 디자인의 애플리케이션 리스트를 제공하고 개편 등의 특정 시점에 변경된 디자인을 반영할 수 있는 새로운 방법을 필요로 한다.

본 연구에서는 국제 규격에 위배되지 않는 멀티 애플리케이션 구조/운용 방법을 제시하고, 본 방송에 대비하여 개발한 실험용 멀티 애플리케이션을 살펴본다. 본 연구에서 제안한 멀티 애플리케이션은 단순히 하나 이상의 애플리케이션을 동시에 제공하는 송출 측면에서의 방법과 달리 애플리케이션 측면에서 구현한 것이므로 다양한 수신기에 대하여 동일한 결과로 멀티 애플리케이션 서비스가 가능하도록 한다.

2. 멀티 애플리케이션 개발

가. 멀티 애플리케이션 기능 개요

애플리케이션 측면에서 멀티 애플리케이션 서비스 환경을 제공하기 위한 핵심은 매니저(혹은 루트) 애플리케션이다. 멀티 애플리케이션 서비스를 제공하기 위하여 방송사는 기존에 서비스하던 형태의 일반적인 애플리케이션들을 관리하기 위한 매니저 애플리케이션도 함께 송출한다. 매니저 애플리케이션은 현재 사용 가능한 일반 애플리케이션들에 대한 리스트를 화면에 표시하고 이들의 라이프사이클을 관리

하는 기능을 하는 특수한 형태의 애플리케이션이다. 즉 매니저 애플리케이션은 다른 애플리케이션들을 시작, 종료, 중단, 파괴시키는 등의 역할을 수행한다.

이러한 기능을 가능하게 하기 위해서 방송사에서는 매니저 애플리케이션의 'control code', 'visibility' 값을 'autostart', '00'으로 각각 셋팅하여 송출하고, 일반 애플리케이션에 대한 이들 값을 'present', '11'로 각각 셋팅하여 송출한다. 수신기는 'autostart'인 매니저 애플리케이션을 수신하자마자 바로 실행시킨다. 수신기에 의해 자동 실행된 매니저 애플리케이션은 현재 실행 가능한 'present' 애플리케이션들에 대한 정보(애플리케이션에 대한 이름, ID, priority, visibility, control code)에 대한 정보를 읽어와 그들에 대한 리스트를 화면에 표시하고 관리를 시작한다. 일반 애플리케이션들의 'visibility' 값은 '11'로 셋팅되어 있으므로 스크린을 통해 사용자에게 보여지고, 매니저 애플리케이션은 애플리케이션 리스팅 API(Application Programming Interface)를 통하여 이들을 감지할 수 있다.

AIT(Application Information Table)이 'control code', 'visibiliyt', '애플리케이션 이름', '애플리케이션 ID' 등 현재 사용 가능한 애플리케이션들에 대한 모든 필요한 정보를 수신기에 전달한다. 데이터 인코더가 이러한 값들을 AIT에 셋팅하여 수신기로 보내면 매니저 애플리케이션은 수신기가 제공하는 AppsDatabase, AppAttributes 등의 자바 API를 사용하여 읽어 온다.

그림 1은 매니저 애플리케션의 일반 애플리케이션들에 대한 관리 기능을, 그림 2는 매니저 애플리케이션에 의한 애플리케이션 리스트 화면을 보여 준다. 그림 1, 2에 도시된 바와 매니저 애플리케이션은 서비스 중인 일반 애플리케이션들의 리스트를 화면에 제공하고 일반 애플리케이션에 대한 네비게이션 기능을 제공한다. 사용자는 화면에 제공된 애플리케이션들의 리스트에서 원하는 애플리케이션을 선택하여 자신이 원하는 서비스를 이용하게 된다.

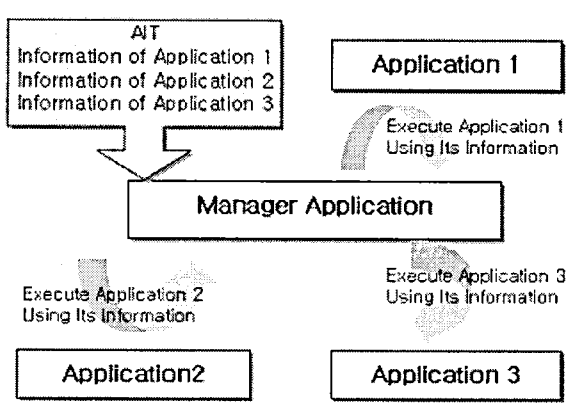

그림 1. 매니저 애플리케이션에 의한 일반 애플리케이션 실행
Fig. 1. Execution of applications by a manager application

그림 2에서 사용자는 리모콘의 상/하 키를 이용하여 네비게이션하다가 원하는 애플리케이션에 포커스가 오면 선택 키를 눌러 원하는 애플리케이션을 실행시킨다. 사용자에 의해 특정 일반 애플리케이션이 선택/실행되면 매니저 애플리케이션은 자신이 점유하고 있던 화면과 포커스를 해당 일반 애플리케이션에게 할당하고 자신을 화면에서 숨긴다. 사용자가 일반 애플리케이션을 사용하다가 특정 메뉴를 이용하여 매니저 애플리케이션을 실행시키거나, 리모콘의 나가기 버튼을 이용하여 애플리케이션에서 빠져 나가는 이벤트가 발생하면 해당 애플리케이션은 이러한 이벤트 정보를 수신기에 전달하고 매니저 애플리케이션은 수신기로부터 이들 이벤트를 받는 즉시 해당 일반 애플리케이션을 화면에서 지운 후 자기 자신을 실행시킨다.

그림 2. 매니저 애플리케이션에 의한 애플리케이션 리스트 표시
Fig. 2. The list of applications displayed by a manager application.

나. 멀티 애플리케이션 구조 및 실행 원리

매니저 애플리케이션은 일반 애플리케이션 각각에 대한 정보를 수신기로부터 읽어와 이들 정보를 이용하여 각각의 애플리케이션들을 구분하고 관리한다. 예를 들어 'organization ID', 'application ID' 등은 일반 애플리케이션 각각에 대하여 고유하게 할당되어 송출되므로, 매니저 애플리케이션은 이들을 수신기로부터 읽어와 애플리케이션들을 구분한다. 그림 3에서 매니저 애플리케이션이 이들 정보를 수신기로부터 획득하는 과정을 나타내었다.

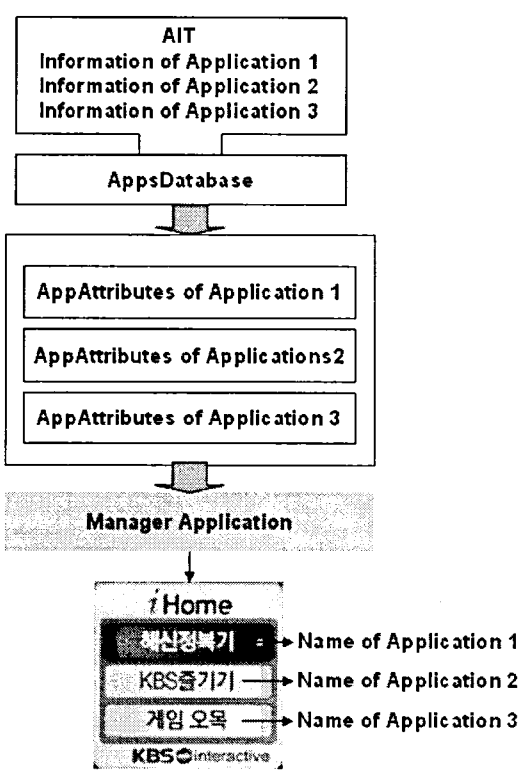

그림 3. 애플리케이션 정보 획득 과정
Fig. 3. Acquisition of the information for applications

그림 3에서 알 수 있듯이, 매니저 애플리케이션은 이들을 획득하기 위하여 수신기가 제공하는 AppAttributes, AppsDatabase 등을 포함하는 org.dvb.application.* 패키지 내의 API들을 이용한다[5, 6]. AppsDatabase는 수신기에서 제공하는 애플리케이션들의 정보를 관리하는 자바 API로, 매니저 애플리케이션은 AIT의 애플리케이션 엔트리들에 대한 정보를 AppsDatabase를 사용하여 얻는다.

아래 소스는 애플리케이션 정보 획득 과정에 대한 일부 소스 코드이다. 매니저 애플리케이션은 AppsDatabase가 제공하는 getAppAttributes(), getAppProxy(AppID)를 통하여 각각의 애플리케이션에 대한 AppAttributes 객체 및 AppProxy 객체를 얻는다. 매니저 애플리케이션은 AppAttributes 객체를 얻은 후 이 객체의 함수들을 사용하여 애플리케이션 정보를 읽어 온다. AppsAttributes 객체는 gerIdentifier(), getName(), getPriority() 등의 함수를 제공하는데, 각각의 애플리케이션에 대한 AppID, 이름, 우선순위를 반환한다.

AppID 객체는 애플리케이션에 대한 고유한 아이디를 표현하기 위한 객체이다. 아래 소스 코드에서 알 수 있듯이, 매니저 애플리케이션은 최종적으로 이 객체의 getAID(), getOID() 함수를 이용하여 애플리케션들을 구분하고 이들 애플리케이션들을 대리할 AppProxy를 얻기 위한 매개변수로 사용한다. 요약하면 매니저 애플리케이션은 AppsDatabase에서 AppAttributes 객체를 얻고 AppAttributes 객체로부터 AppID를 포함한 정보를 얻어 이들 정보를 이용하여 화면에 애플리케이션 리스트를 표시하고 관리를 한다.

```
if(app_database == null)
{
  app_database = AppsDatabase.getAppsDatabase();
}
CurrentServiceFilter cur_filter = new CurrentServiceFilter();
if((app_database != null) && (app_database.size() != 0))
{
  Enumeration attributes = app_databse.getAppAttributes(cur_fil
  ter);
  if(attribures != null)
  {
    while(attributes.hasMoreElemetns())
    {
      info = (AppAttributes)attributes.nextElement();
      if(info != null && info.getIsServiceBound())
      {
        app_id = info.getIdentifier();
        aid = app_id.getAID();
        oid = app_id.getOID();
        app_proxy = app_database.getAppProxy(app_id);
        try{
          app_name = new String(info.getName().getBytes("UTF
          -8", "UTF-8");
        }catch(Exception e){
          e.printStackTrace();
        }
```

화면에 표시된 애플리케이션 리스트 중 하나를 시청자가 선택하여 실행할 경우, 매니저 애플리케이션은 해당 애플리케이션이 시작되도록 한다. 그림 4에서 세 개의 일반 애플리케이션 중 'application 1'을 사용자가 선택/실행할 경우에 해당 애플리케이션이 실행되는 과정을 나타내었다. 그림 4에서와 같이 매니저 애플리케이션은 사용자가 선택한 애플리케이션을 실행하기 위하여 AppProxy라는 객체를 사용한다.

위 소스 코드에서 살펴본 바와 같이, AppsDatabase, AppAttributes, AppID를 순서대로 얻어 온 후 최종적으로 AppsDatabase.getAppProxy(AppID)를 통하여 AppProxy를 미리 얻어 놓고, 사용자가 특정 애플리케션을 실행시키면 즉시 AppProxy.start()를 호출하여 해당 애플리케이션을 시작시킨다. AppProxy는 특정 애플리케이션의 동작 상태를 제어하는 proxy 역할을 하도록 수신기에서 제공하는 애플리케이션 대리자(Proxy)이다.

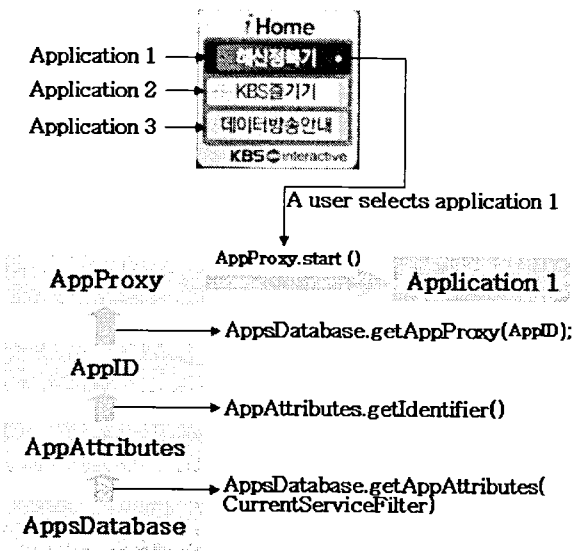

그림 4. 애플리케이션 시작 과정

Fig. 4. Steps for starting an application

AppProxy는 AIT에 시그널 되는 각 애플리케이션 엔트리와 일대일 대응 관계를 가진다. AppProxy는 해당 애플리케이션을 시작, 종료, 일시 중지시키는 등의 함수를 제공한다.

애플리케이션은 자신이 실행되고 있는 도중에 사용자가 리모콘의 실행 중지 버튼을 누를 경우 해당 정보를 수신기에 보내고, 매니저 애플리케이션은 수신기로부터 중지 정보를 전달받아 해당 애플리케이션의 AppProxy를 얻어 AppProxy의 stop() 함수를 호출하여 해당 애플리케이션이 중지되도록 한다. 이를 위하여 매니저 애플리케이션은 AppStateChangeEventListener를 구현하여, 애플리케이션 상태가 변하면 수신기가 발생시키는 AppStateChangeEvent를 처리한다.

송출 시스템에서 현재 서비스 중인 애플리케이션을 중단시키거나 새로운 애플리케이션을 송출할 경우, 매니저 애플리케이션은 이들 이벤트를 처리한다. 이를 위하여 매니저 애플리케이션은 AppsDatabaseEventListener를 구현하여 AppsDatabase 내용이 변할 경우 수신기가 발생시키는 AppsDatabaseEvent를 처리한다. 현재 서비스 중인 애플리케이션 엔트리 중 하나가 서비스 중단되거나 변경될 때, 엔트리가 새로 추가될 때 수신기는 AppsDatabaseEvent를 발생시킨다. 매니저 애플리케이션은 AppsDatabaseEvent가 발생하면 자신이 가지고 있는 애플리케이션들에 대한 모든 정보 및 화면에 표시한 애플리케이션 리스트를 갱신한다.

다. 실험용 멀티 애플리케이션 개발 예

2005년 KBS 방송기술연구팀에서는 실험용 멀티 애플리케이션

을 개발하여 멀티 애플리케이션 고유 기능에 대한 동작 실험 뿐만 아니라, 송출 시스템 및 수신기의 멀티 애플리케이션 지원 기능에 대한 정합 실험을 실시하였다. 매니저 애플리케이션 및 게임 2종을 포함하여 아래의 총 6종류의 다양한 콘텐츠를 개발하여, 멀티 애플리케이션 기능 뿐만 아니라 양방향 기능, 리턴 서버를 이용한 데이터 송출 등 다양한 기능을 테스트할 수 있었다.

- '매니저 애플리케이션'
  - 멀티 애플리케이션 송출 시스템을 통해 두 개 이상의 애플리케이션을 송출하고, 제어하는 기능 실험
  - 복수 개의 애플리케이션을 시청자가 선택적으로 접근하는 기능, 네비게이션 기능, 제어 기능 등 멀티 애플리케이션 고유 기능 실험
  - 수신기의 멀티 애플리케이션 지원 기능 정합 실험
- '해신 정복기' 애플리케이션 개발
  - 퀴즈 참여, 상품 구매 등을 통한 송출 시스템 및 수신기에 대한 양방향 기능을 점검하기 위한 애플리케이션 개발
  - 수신기 시간 정보를 이용하여 계산한 이용시간 정보 등을 이용한 시청자 반응분석 시스템 검증 실험
- 단방향 애플리케이션 개발: '오목'/'퍼즐' 게임, '데이터방송안내'
- 'KBS 즐기기' 애플리케이션 개발
  - 리턴서버의 대용량 데이터를 리턴채널로 전송하는 기능 실험
  - 대용량 데이터 전송의 효과를 반영하기 위한 풍부한 이미지와 텍스트로 구성된 메뉴 제공

그림 5는 애플리케이션 리스트 화면이다.

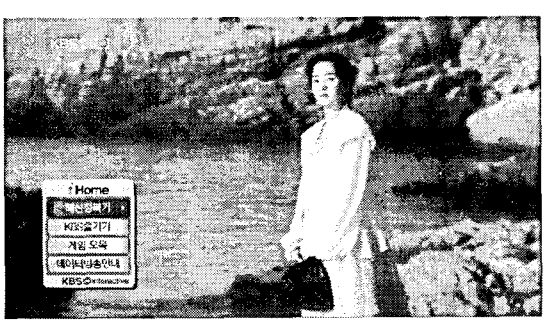

그림 5. 애플리케이션 리스트
Fig. 5. The list of applications

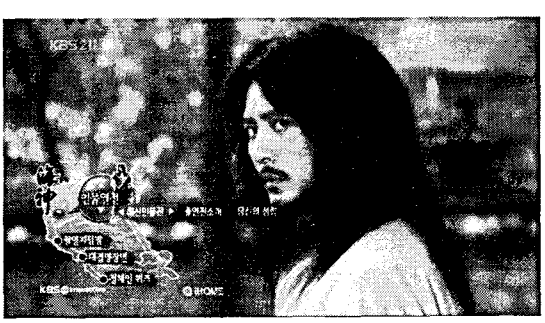

그림 6. '해신 정복기' 메인 메뉴
Fig. 6. The main menu of '해신 정복기'

애플리케이션 리스트 화면에서 '해신 정복기'를 선택하면 그림 6과 같이 '해신 정복기' 애플리케이션이 실행된다.

3. 결론

본 논문에서는 데이터방송시 복수의 애플리케이션을 제공할 수 있는 멀티 애플리케이션 개발에 대하여 다루었다. 특히 본 논문에서 제안한 멀티 애플리케이션은 단순히 일반 애플리케이션을 여러 개 송출하는 것이 아니라, 애플리케이션 측면에서 서비스를 제공하므로 사용자의 수신기 종류에 관계없이 멀티 애플리케이션 서비스를 제공할 수 있다.

제안한 멀티 애플리케이션이 본 방송에서 실시되면 사용자는 자신의 기호에 맞는 다양한 서비스를 즐길 수 있고 방송사는 편성의 자유를 가질 수 있을 것이다. 추후 본 방송에 적용되기 위해서는 매니저 애플리케이션의 지정자 화면, 애플리케이션 리스트 표현 등의 사용에 대한 심도 있는 고려가 필요하다.

참고 문헌

[1] ATSC Standard A/101: Advanced Common Application Platform(ACAP), 2 August 2005.

[2] ATSC homepage: http://www.atsc.org/

[3] Jeong-Deok Kim, Sangjoo Lee, and Hyun-Soon Kim, "Developing a Fully Interactive TV System and Applications in Compliance with ATSC ACAP," Conference Proceedings, NAB 2005 Broadcast Engineering, 2005.

[4] Digital Video Broadcasting(DVB), Globally Executable MHP version 1.0.2, ETSI TS 102 819 v.1.3.1.

[5] Digital Video Broadcasting(DVB), Multimedia Home Platform(MHP) version 1.0.3, ETSI TS 101 812 v.1.3.1.

[6] Digital Video Broadcasting(DVB), Multimedia Home Platform(MHP) version 1.1.1, ETSI TS 101 812 v.1.2.1.