

# 데이터 방송 위젯 애플리케이션 개발

이차원<sup>1</sup>, 정문열<sup>2</sup>  
서강대학교 영상대학원 미디어공학과 디지털 방송 연구실<sup>1,2</sup>  
{ah2dius<sup>1</sup>, moon<sup>2</sup>}@sogang.ac.kr.

## Development of Widget Application for Data Broadcasting

Cha Won Lee<sup>1</sup>, Ryul Moon Jung<sup>2</sup>  
Digital Broadcasting Lab, Graduate School of Media Communications,  
Sogang University<sup>1, 2</sup>

### 요약

기존의 정보 제공형 서비스에 시청자가 접근하기 위해서는 TV 시청을 중단하고 여러 단계를 거쳐서 이동해야 하고, 접근하는 시간이 시청자가 원하는 시간보다 오래 걸리는 단점이 있다. 본 논문에서는 이러한 접근성을 극복하기 위하여 데이터방송 위젯 애플리케이션을 제안한다.

본고에서는 실제 방송에 적용하기 위하여 개발한 실험용 위젯 애플리케이션을 중심으로 살펴보고자 한다. 위젯 애플리케이션으로 적용시킬 수 있는 정보는 무한하므로 위젯의 구조 또한 쉽게 응용하여 만들 수 있도록 설계하였다. 그리고, 화면상에 하나의 위젯만 떠있는 것이 아니라 다중의 위젯이 떠있는 상태를 관리할 수 있는 위젯 관리 애플리케이션도 살펴본다. 특정 플랫폼에 국한하지 않고 모든 플랫폼에 적용될 수 있도록 개발하였다

Keyword : Data Broadcasting, xlet, widget

## 1. 서론

디지털 케이블 방송의 시작으로 본격적인 데이터방송 콘텐츠 시장이 열리게 되었다. 이를 바탕으로 데이터방송 사업자들은 다양한 콘텐츠를 선보이고 있다. 하지만, 의욕적으로 선보인 콘텐츠들이 시청자들에게 새롭게 받아 들이지 못하고 있다. 여러 가지 이유가 있겠지만 시청자들의 인식이 아직도 단순히 TV 는 시청만 하는 용도로 생각한다는 점과 콘텐츠에 접근하기 힘든 점이다. 간단한 날씨 정보를 보기 위해서는 시청중인 영상을 멈추고 몇 단계를 거쳐야 확인을 할 수 있게 되어있다. 이러한 접근성의 한계를 극복하기 위해 본고에서는 데이터방송 위젯 애플리케이션을 제안한다. 위젯 애플리케이션이란 TV 시청 중에도 항상 시청자가 선택한 정보가 영상 위에 떠있게 하는 애플리케이션을 말한다. 위젯은 정보의 접근을 용이하게 하고, 영상 위에 표현되는 위젯은 화면

크기가 작고 투명 처리가 되어 시청자에게 거부감이 들지 않는 장점이 있다. 그리고, 위젯에는 간략한 정보만을 표현하여 세부 정보를 알고 싶을 때에는 시청자가 그 위젯을 선택하여 정보 제공 서비스에 접속할 수 있게 된다. 기본적인 정보 제공형 서비스로서의 시간, 날씨, 뉴스 등을 적용할 수 있다. 더불어, 방송사의 이벤트나 광고를 노출시켜 다양한 비즈니스 모델을 적용시킬 수 있을 것이다.

본고에서는 실제 방송에 적용하기 위해 개발한 실험용 위젯 애플리케이션을 중심으로 살펴보고자 한다. 위젯의 확장성을 고려하여 누구나 쉽게 제작할 수 있도록 설계하였다. 또한, 위젯은 단순히 하나의 애플리케이션이 아니라 여러 가지 정보를 가진 복수의 위젯이 있을 수 있으므로 이를 관리하기 위한 위젯 관리 애플리케이션도 살펴 보도록 하겠다.

## 2. 위젯

### 2-1 위젯의 의미

일반적인 의미의 widget(위젯)은 약간의 기계적인 특질을 가지고 있고, 상대적으로 적은 크기이며, 그것이 특별한 이름을 가지고 있지 않을 때, 또 그것의 이름을 기억할 수 없거나, 일반적으로 잘 알려지지 않는 어떤 객체들의 부류에 관해 이야기할 때, 어떠한 분리된 객체를 가리키는데 사용할 수 있는 용어이다.

컴퓨터에서의 위젯은 정보를 나타내거나, 운영체제 및 응용프로그램 등과 상호 작용하려는 사용자를 위해 특유한 방법을 제공하는 GUI 구성요소이다. 위젯에는 아이콘, 풀다운 메뉴, 푸시 버튼, 선택 상자, 체크 박스, 스크롤 바, 창, 토글 단추, 폼 그리고 정보를 나타내고 받아들이고 사용자 요구에 응답하기 위한 많은 다른 구조들을 포함한다. 하지만, 본고에서의 의미는 작은 애플리케이션을 뜻한다. PC 환경에서는 이미 적용되어 사용중인 위젯은 사용자가 자주 찾는 인터넷의 정보를 웹브라우저 구동 없이 바로 바탕화면에서 별도의 창을 통해 즉각 얻을 수 있다. 그림 1에서 보듯이 날씨, 달력, 시계, 일정, 계산기, 심지어 작은 게임에 이르기까지 다양한 응용 범위가 있다.



그림 1. PC 환경에서의 위젯

### 2-2 데이터 방송 위젯

데이터 방송에서 제공되는 애플리케이션 중 위젯과 비슷한 형태의 애플리케이션은 미니 EPG이다. 채널 변경이나 ‘+’ 버튼을 선택하면 화면

하단에 프로그램 제목과 채널 정보를 보여주는 그림 2와 같은 미니 EPG가 나타나게 된다. 반투명므로 TV 화면에 나타나지만 항상 떠 있지 않고 몇 초 후에 화면에서 사라진다.



그림 2. 미니 EPG

미니 EPG는 간단한 정보를 제공한다는 점에서 위젯의 특성에 부합되지만 위젯의 위치변경이 불가능한 점과 항상 떠있지 않는 점에서 완전한 위젯이라고는 할 수 없을 것이다.

데이터 방송 위젯에서는 독립형 서비스와 연결의 특징이 있다. 위젯은 간단한 정보만을 제공하는 것이 아니라 그에 해당하는 독립형 정보 서비스와 연동하여 해당 위젯에서 곧바로 정보 서비스로 이동하여 세부적인 정보를 받을 수 있다. 이와 같은 특징은 독립형 정보 서비스의 접근성을 해결해 줄 수 있는 점이다.

복수의 위젯이 TV 화면에 떠있을 경우 시청자의 시청화면을 막아버리는 수가 있다. 이러한 점을 해결하기 위해 각각의 위젯은 반투명 처리를 하여 최소한의 시청을 보장하며 점차 디스플레이가 커짐에 따라 위젯들을 화면 모서리에 위치시켜도 시청에 큰 무리가 따르지 않을 것이다.

### 2-3 데이터 방송 위젯에서 고려해야 할 점

데이터 방송에서 돌아갈 애플리케이션을 Xlet이라 하는데, 방송 스트림에 포함되어 다운로드된다. Xlet의 프로그램의 구조나 life cycle은 인터넷 기반의 applet과 매우 유사하다. 이러한 Xlet을 통하여 위젯 애플리케이션을 만들 때 가장 유의할

점은 리소스 관리이다. 컴퓨터 환경보다 열악한 리소스 상태에서 여러 개의 위젯을 띄우기 위해서는 리소스와 위젯을 관리할 매니저 어플리케이션이 필요하다. 또한, 그래픽 자원이 많은 리소스를 점유하므로 위젯 작성시 주의해야 한다. 그리고, 날씨, 주식정보 등 과 같이 항상 외부로부터 정보를 가지고 와야 하는 것들은 리턴 채널을 이용하던지 방송국의 데이터서버에서 주기적으로 최신 데이터를 전송해야 하는 점도 유의해야 할 것이다.

### 3. 위젯 어플리케이션 개발

#### 3-1 위젯 어플리케이션 설계

위젯 어플리케이션 설계에 가장 중심을 둔 요소는 확장성이다. Xlet 를 잘 모르더라도 기본적인 프레임워크를 제공하여 쉽게 개발할 수 있도록 설계하였다.

프레임워크에는 크게 두 부분으로 나뉜다. 기본적인 Xlet , 그래픽 환경 설정 부분과 실제 위젯 컴포넌트 작성 부분이다. 첫 번째 Xlet 환경 설정 부분을 살펴보겠다.

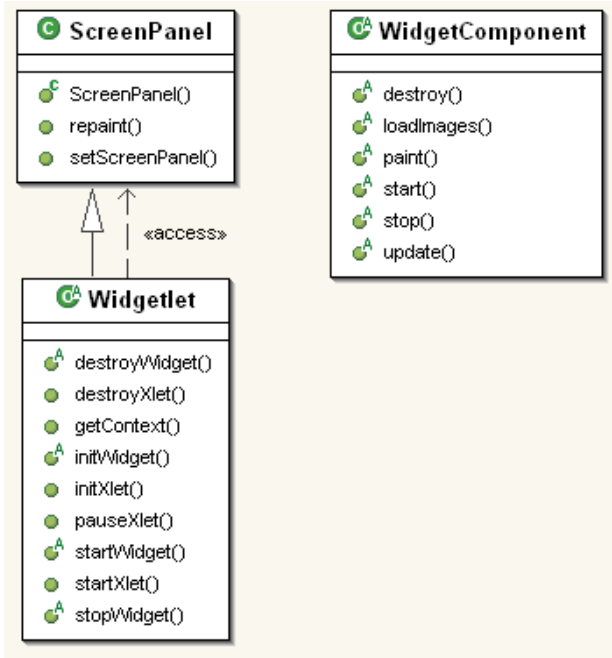


그림 3. 프레임워크 클래스 다이어그램

ScreenPanel Class 는 기본적인 그래픽 환경을 잡아주게 된다. HContainer 를 상속받아 HScene 를 획득하여 화면에 컴포넌트들을 올려놓을 수 있도록 환경 설정을 한다.

Widgetlet Class 는 ScreenPanel Class 를 상속받아 Xlet 에서 사용할 수 있도록 하였으며 Xlet 인터페이스를 구현하였고 initWidget(), startWidget(), stopWidget(), destroyWidget() 의 추상 메서드를 생성 하였다. Widgetlet Class 를 상속받아 작성하는 위젯 어플리케이션에서 추상 메서드를 작성하도록 하여 자동적으로 initXlet()를 하는 효과와 개발자에게는 Xlet 을 은닉하여 신경을 쓰지 않아도 되는 특징이 있다. 위젯 개발자는 단지 컴포넌트를 만들고 그 컴포넌트를 생성하여 initWidget() 내에서 add()를 하게 되면 Xlet 으로 작성된 위젯이 완성 되는 것이다.

프레임워크의 두번째 부분인 위젯 컴포넌트를 살펴보겠다. WidgetComponent 추상 클래스는 컴포넌트를 구성하는 가장 기본적인 메서드로 구성되어 있다. Widgetlet 과 연결 고리인 start(), stop(), destroy() 그리고, 이미지 객체 로딩, 컴포넌트를 그려줄 update(Graphics g), paint(Graphics g) 로 구성되어 있다. 만들고 싶은 위젯 컴포넌트 클래스에서는 WidgetComponent 추상 클래스를 상속받으면 기본적인 컴포넌트가 생성이 된다.

위에서 살펴본 두 개의 프레임워크를 이용하면 쉽게 기본적인 위젯 어플리케이션을 개발할 수 있게 된다. 특히 첫 번째 프레임워크는 일부분을 약간 수정만 하면 위젯 어플리케이션 뿐만 아니라 다른 어플리케이션 제작에도 유용히 쓰일 것이다.

#### 3-2 위젯 어플리케이션 개발

위에서 설계된 위젯 프레임워크를 이용하여 디지털 시계 위젯을 개발하였다. 기본적인 Xlet 설정과 그래픽 설정은 프레임워크를 기반으로 구성하였다. 위젯 어플리케이션에서 핵심적인 부분은 컴포넌트 부분을 개발하는 것이다.

DigitalClockComponent Class 는 WidgetComponent 추상 클래스를 상속받아 추상 메서드를 구현하여 기본적인 컴포넌트의 틀을 갖췄다. 시계의 특성상 실시간으로 값이 바뀌어야 하므로 내부 클래스로 타이머를 스레드로 돌려서 항상 시간 값을 가져오도록 했다. start() 에서 스레드를 생성하고 시작하도록 하였다. stop() 에서는 스레드를 중지하도록

하였다. `destroy()`에서는 쓰레드를 중지하고 로딩되어있던 그래픽 객체를 해제시키도록 했다. `loadImages()`는 위젯에 보여질 이미지를 `MediaTracker()`를 이용하여 메모리에 로딩시키도록 했다. `updateTime()`이라는 메서드를 만들어서 시간 값들을 시분초로 분리하여 변환하고 이를 `paint()`에게 넘겨줘서 바뀐 시간 값이 그려지도록 하였다. `KeyListener` 처리를 위해 새로운 클래스를 생성하여 `DigitalClockComponent`를 상속하고 `KeyListener`를 구현하였다. 컴포넌트와 `KeyListener`를 다른 클래스로 구별한 이유는 MVC 모델링을 적용했기 때문이다. 그림 4에서 이러한 클래스 구조를 나타내고 있다.

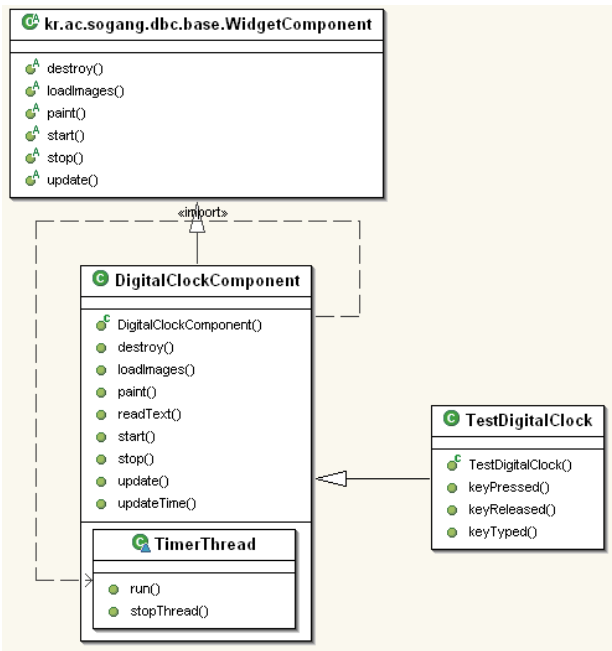


그림 4. 디지털 시계 클래스 다이어그램

이렇게 개발된 위젯을 실제 방송 환경에서 테스트를 해보았다. MHP와 OCAP 환경에서 테스트를 해보니 그림 5와 같이 화면에 나타났다.



그림 5. 위젯 테스트 화면

#### 4. 위젯 관리 애플리케이션 개발

##### 4-1 위젯 관리 애플리케이션 기능

매니저 애플리케이션은 복수개의 위젯 애플리케이션을 관리하는데 목적이 있다. 시청자에게 현재 사용 가능한 위젯 애플리케이션 리스트를 보여 주거나 그들간의 라이프사이클을 관리하는 기능을 가진다. 이러한 기능을 가능하기 위해서는 송출 시스템에서 애플리케이션을 보낼 때 설정할 값들이 정의되어 있다. 매니저 애플리케이션은 `Control Code` 값을 `Auto Start`로 설정하고, `Visibility` 값을 `Both(API/User)`로 설정한다. 그리고, 위젯 애플리케이션은 `Control Code` 값을 `Present`로 설정하고, `Visibility` 값을 `API`로 설정한다. 송출 시스템에서의 값으로 설정하여 보내게 되면 화면에는 매니저 애플리케이션만 보이게 되고 다른 위젯 애플리케이션의 리스트를 획득할 수 있게 된다.

그림 6은 매니저 애플리케이션에 의해 실행되는 위젯을 관리하는 기능을 보여준다. `AIT(Application Information Table)`에 저장된 애플리케이션의 정보를 매니저 애플리케이션에서 획득하여 위젯 애플리케이션의 리스트를 보여주게 된다. 시청자가 제공된 리스트에서 원하는 위젯을 선택하게 되면 그 위젯이 실행이 된다.

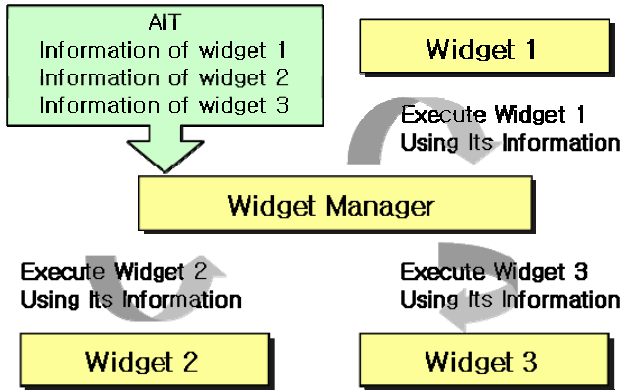


그림 6. 매니저 애플리케이션에 의한 위젯 실행

#### 4-2 위젯 관리 애플리케이션 설계

위젯 관리 애플리케이션도 Xlet 이므로 위젯 애플리케이션 작성에 사용했던 프레임워크를 적용하였다. 사용자에게 보여줄 리스트 화면도 컴포넌트로 구성하였다. 일반 위젯 애플리케이션과 다른 점은 AIT 테이블로부터 애플리케이션에 대한 정보를 가져오는 클래스가 추가된 것이다.

그림 7 은 애플리케이션 정보를 가져오는 AppsManager Class 를 나타낸 것이다. getNames() 메서드를 통하여 AIT 에 있는 애플리케이션 정보 중 이름을 String 배열로 가져오게 된다. 이 정보를 바탕으로 리스트를 구성하게 되며 시청자가 리스트 중 특정 위젯 애플리케이션을 선택했을 경우 searchAppID(String name) 메서드를 이용하여 해당 AppID 를 가져오게 된다. 그 AppID 를 바탕으로 getAppProxy(AppID app) 메서드를 이용하여 AppProxy 를 가져올 수 있게 된다. AppProxy.start() 로 선택한 위젯 애플리케이션이 실행이 되게 된다.

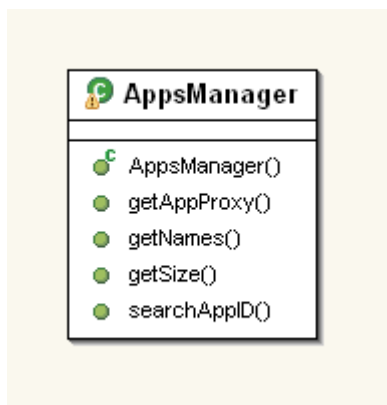


그림 7. AppsManager 클래스 다이어그램

AppProxy 는 특정 애플리케이션의 동작 상태를 제어하는 proxy 역할을 하도록 수신기에서 제공하는 애플리케이션 대리자(proxy)이다. 또한, 시청자에 의한 사용 중지가 되었을 경우 AppStateChangeEvent 처리를 하여 AppProxy.stop() 을 실행하도록 한다.

#### 4-3 위젯 관리 애플리케이션 개발

위에서 설계된 AppsManager Class 와 위젯 프레임워크를 이용하여 매니저 애플리케이션을 개발하였다. 위젯 컴포넌트는 AppsManager Class 에 접근하여 리스트 정보를 획득하고 리스트를 화면에 그려주는 기능을 한다.

실제 방송 환경에서 테스트를 하기 위해 송출시스템에서 애플리케이션의 Control Code, Visibility 값을 설정하여 송출하여 확인 해보았다.

그림 8 에서 보듯이 위젯 매니저 애플리케이션을 통해 전자 시계 위젯이 화면에 나타난 것을 알 수 있다.



그림 8. 위젯 매니저 테스트 화면

#### 4-4 위젯 매니저 애플리케이션 평가

위젯 매니저 애플리케이션의 중심인 위젯 리스트 관리는 AppsDatabase 클래스를 이용하여 쉽게 가능하였다. 하지만, 리소스 관리 부분은 아직 구현이 되어 있지 않아 복수개의 위젯이 뜰 경우 리소스가 부족하여 실행이 되지 않을 수 있다. 또한, 채널을 변경해도 항상 떠있어야 하므로 OCAP 환

경에서 unbound application 으로 설정해야 한다. 실험 환경에서는 아직 구축이 되지 않아 테스트를 해보지 못했다. 그리고, 송출 시스템에서 애플리케이션 설정을 할 때 이름을 한글로 지정해주면 AIT 테이블에서 이름 값을 가져올 때 한글이 안 나오는 현상이 발생했다. 'UTF-8' 로 인코딩 타입을 변환 했지만 제대로 나오지 않아 송출 시스템의 문제인지 셋톱박스의 문제인지는 확인하지 못했다. 위젯 매니저 애플리케이션 자체의 리소스를 줄이기 위해 배경 이미지를 빼놓고는 리스트와 네비게이션 환경은 Graphics Class 의 draw 메서드를 이용하여 구성하였다.

## 5. 결 론

본 고에서 데이터방송 위젯 애플리케이션의 의미와 위젯의 설계 및 개발 그리고, 위젯들을 관리하는 위젯 관리 애플리케이션의 설계 및 개발에 대해 살펴보았다.

실제 방송 환경에서 손쉽게 개발할 수 있도록 위젯 프레임워크를 제공하였고 이를 바탕으로 전자 시계 위젯을 개발하였다. 단순한 시계뿐만 아니라 날씨, 일정관리, 주식시세 등등 다양한 아이디어가 결합한 위젯이 나오리라 여긴다. 방송사 입장에서는 자사 프로그램을 홍보할 수 있는 창으로 만들 수도 있고 시청자 입장에서는 기본적인 정보를 TV 시청하다가 자연스럽게 정보를 받아들일 수 있는 장점이 있을 것이다.

현재 다양한 데이터방송 콘텐츠들이 개발되면서 점차 시청자들의 이용률이 높아가고 있지만, 만족도에서는 낮은 편이다. 이러한 시청자들의 눈높이를 맞추기 위해서는 TV 환경에 걸맞는 다양한 콘텐츠들이 개발되어야 할 것이다.

## 참고문헌

- [1] 정문열, 백두원, “ 연동형 데이터방송 애플리케이션 구조”, 방송공학회 논문지, 제 9 권 1 호, 2004
- [2] 김현순, “ 멀티 애플리케이션 개발”, 방송공학회 학술대회, 2005

- [3] Chengyuan Peng, Petri Vuorimaa, “ Development of Java User Interface For Digital Television”, HCI, 1999
- [4] 김동원, “ 데이터방송 서비스를 위한 화면디자인 연구”, 국민대학교 테크노디자인전문대학원, 2001
- [5] Multimedia Home Platform 1.0.1 – DVB Blue book A057 Rev.1, European Broadcasting Union, 2001/1.