

# 개방형 IPTV 서비스 개발을 위한 서비스 생성환경 제공방법

유초롱\*, 이현진\*, 조기성\*

\*한국전자통신연구원

e-mail : {crryu, petrus, chokis}@etri.re.kr

## A Method of Service Creation Environment for open IPTV Service development

Cho-Rong Yu\*, Hyun-Jin Lee\*, Kee-Seong Cho\*

\*Electronics and Telecommunications Research Institute

### 요 약

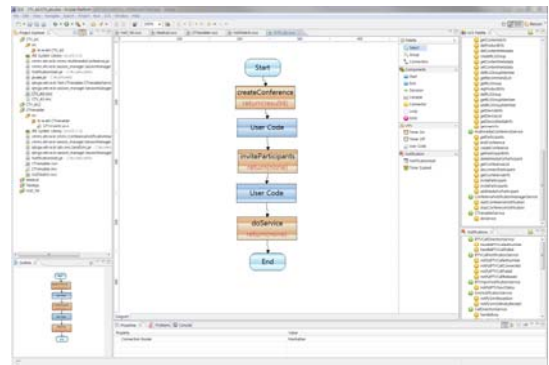
본 논문은 개방형 IPTV 서비스를 위한 서비스 생성도구에 대한 것으로, 웹 서비스 기반의 컴포넌트를 기본단위로 GUI(Graphic User Interface) 상에서 각각의 컴포넌트를 조합하여 IPTV 융합 서비스를 생성하는 서비스 생성환경 제공방법에 대해 소개한다. 개방형 IPTV 플랫폼에서는 방송제어, 통신제어, 웹 콘텐츠 제어, 정보 제어 기능 등을 웹 서비스 기반의 Open API 로 제공하고 있으며, 서비스 생성도구는 개방형 IPTV 플랫폼에서 제공하는 Open API 를 컴포넌트로 하여 컴포넌트들을 조합하고 연결하기 위한 GUI 환경인 그래픽 에디터, 그래픽 에디터에서 완성한 서비스 로직에 대한 서비스 프로그램 코드를 생성하기 위한 코드 생성기, 서비스 프로그램을 컴파일하고, 컴파일이 완료된 서비스 프로그램을 배포하는 서비스 디플로이어로 구성된다.

### 1. 서론

SCE(Service Creation Environment)는 과거 지능망에서부터 사용되던 개념으로 SCP(Service Control Point) 상의 서비스를 생성하기 위한 개발 환경이다. 개방형 IPTV 플랫폼은 IPTV 서비스 제공을 위해 방송서비스 제어, 통신서비스 제어, 웹 콘텐츠 제어, 정보 제어, 융합서비스 제어와 같은 기능을 Open API 형태로 제공하고, 이를 통해 IPTV 서비스 사업자가 다양한 IPTV 융합 서비스를 개발할 수 있도록 한다. 개방형 IPTV 플랫폼 서비스 생성도구는 개방형 IPTV 플랫폼에서 제공하는 다양한 API 들을 활용하여 융복합 API 를 생성하거나 원격 교육, 원격 의료와 같은 응용 서비스를 GUI 를 통해 쉽게 개발할 수 있도록 한다.

### 2. 개방형 IPTV 서비스 생성도구

개방형 IPTV 서비스 생성도구는 개방형 IPTV 플랫폼에서 제공하는 Open API 들을 컴포넌트화하고, 캔버스에서 각각의 컴포넌트를 조합하고 연결하여 서비스 로직을 생성하는 그래픽 에디터, 그래픽 에디터에서 생성한 서비스 로직에 대한 서비스 프로그램 코드를 생성하는 코드 생성기, 서비스 프로그램을 컴파일하고, 서비스 프로그램을 응용 서버 또는 개방형 IPTV 플랫폼과 같은 웹 서버에 배포하는 서비스 디플로이어로 구성된다.



(그림 1) 개방형 IPTV 서비스 생성도구

#### 2.1 그래픽 에디터

서비스 생성을 위한 그래픽 에디터는 Java Eclipse 상의 플러그인 형태로 개발되었다.

그래픽 에디터는 크게 캔버스와 팔레트로 구성된다. 팔레트는 서비스 생성도구에서 제공 가능한 컴포넌트들을 보여준다. 이때, 개방형 IPTV 플랫폼에서 제공하는 웹 서비스 기반 Open API 들이 컴포넌트로 포함된다. 이들 컴포넌트들 이외에 서비스 로직을 위한 서비스 컴포넌트들과 유틸리티 컴포넌트들 역시 제공된다.

서비스 컴포넌트에는 서비스 로직의 처음을 나타내는 'Start', 서비스 로직의 마지막으로 나타내는 'End', 컴

포넌트들의 중복수행을 위한 'Loop', 서비스 로직에서 특정 조건에 대해 분기 조건을 정의할 수 있는 'Decision', 서비스 로직에서 사용할 변수를 정의하는 'Variable' 등이 포함된다.

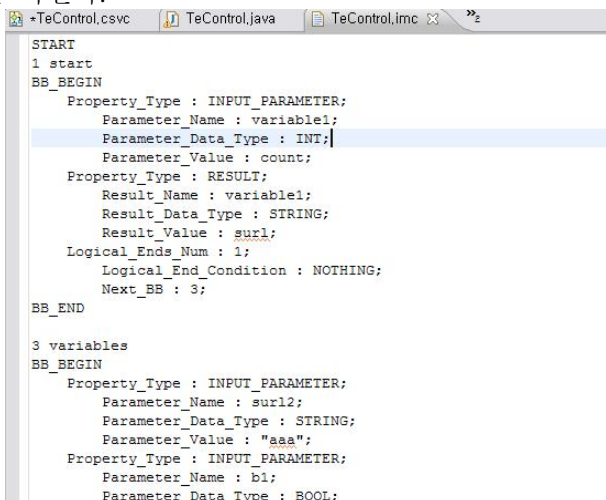
유틸리티 컴포넌트는 서비스 로직이나 개방형 IPTV 플랫폼으로부터 전달되는 이벤트를 수신하기 위한 'NotificationWait', 타이머 기능을 설정/해제할 수 있는 'TimerOn/TimerOff', 타이머가 만료되었음을 공지하는 'TimerExpired', 서비스 개발자가 직접 서비스 로직 내에 추가하고 싶은 프로그램 코드를 작성하여 추가할 수 있는 'UserCode' 등이 포함된다.

서비스 컴포넌트와 유틸리티 컴포넌트는 그래픽 에디터에 포함되어 개발된 것으로 개방형 IPTV 플랫폼에서 제공하는 기능에 해당하는 컴포넌트와는 동작 방식이 상이하다.

개방형 IPTV 플랫폼의 Open API 는 웹 서비스로 그래픽 에디터에서 각 Open API 에 해당하는 WSDL 를 읽어 들이면 Open API 를 사용할 수 있는 클라이언트 스텝을 자동으로 생성하고, WSDL 에 정의되어 있는 입출력 인터페이스를 분석하여 그래픽 에디터의 팔레트에 직접 컴포넌트로 추가할 수 있는 기능을 제공한다. 이는 개방형 IPTV 플랫폼 Open API 뿐만 아니라 웹 서비스로 개발된 다양한 API 들에 대하여 그래픽 에디터에 컴포넌트로 추가하여 활용할 수 있도록 확장성을 고려하여 설계하였다. 이밖에 그래픽 에디터는 에디터의 기본 기능을 모두 제공하고 있으며, 캔버스에 작성한 서비스 로직을 저장하여 중간코드 형태로 저장하는 기능을 추가적으로 제공한다.

## 2.2 코드 생성기

그래픽 에디터에서 생성한 서비스 로직은 GUI 형태의 서비스 로직과 중간코드로 저장된다. 중간코드는 서비스 로직에 대하여 실제 프로그래밍 언어로 변환하기 위한 것으로, 아래와 같은 텍스트 기반의 형식을 가진다.



```

START
1 start
BB_BEGIN
  Property_Type : INPUT_PARAMETER;
  Parameter_Name : variable1;
  Parameter_Data_Type : INT;
  Parameter_Value : count;
  Property_Type : RESULT;
  Result_Name : variable1;
  Result_Data_Type : STRING;
  Result_Value : url;
  Logical_Ends_Num : 1;
  Logical_End_Condition : NOTHING;
  Next_BB : 3;
BB_END

3 variables
BB_BEGIN
  Property_Type : INPUT_PARAMETER;
  Parameter_Name : url2;
  Parameter_Data_Type : STRING;
  Parameter_Value : "aaa";
  Property_Type : INPUT_PARAMETER;
  Parameter_Name : b1;
  Parameter_Data_Type : BOOL;

```

(그림 2) 서비스 로직 중간코드

(그림 2)에서 보는 바와 같이 중간코드는 각 컴포넌트의 입출력 파라미터 및 연결된 컴포넌트들에 대한 정

보를 포함하고 있다. 서비스 로직을 구성하는 데 있어, 각 컴포넌트들 사이의 구동순서는 매우 중요한 의미를 가진다. 상위 컴포넌트로부터 전달된 출력 파라미터는 연결된 하위 컴포넌트의 입력 데이터로 활용될 수 있으며 이는 구동 순서에 제약이 있는 컴포넌트들 사이에 효과적으로 활용된다. 예를 들어, 양 사용자간에 IPTV Call 연결을 요청하는 'makeCall' 컴포넌트에서, IPTV Call 연결이 반드시 먼저 선행되고 'endCall' 컴포넌트를 통해 연결된 IPTV Call 을 종료할 수 있다. 이때, 'makeCall' 컴포넌트는 연결된 IPTV Call 에 대한 callidentifier 값을 반환한다. 이 callidentifier 값은 'endCall' 컴포넌트의 입력 파라미터 값으로 설정될 수 있다.

코드 생성기는 중간코드를 읽어 들여 분석하고, 각 컴포넌트 별로 프로그램 변환에 필요한 정보를 이용하여 Java 언어로 서비스 프로그램을 변환하게 된다. 웹 서비스 Open API 단위의 컴포넌트의 경우, Open API 를 컴포넌트 단위화하기 위해 WSDL 을 읽어 들여 서비스 생성도구 사이트에 클라이언트 스텝(Client Stub)을 생성하여 Open API 호출이 가능하도록 해야 한다.

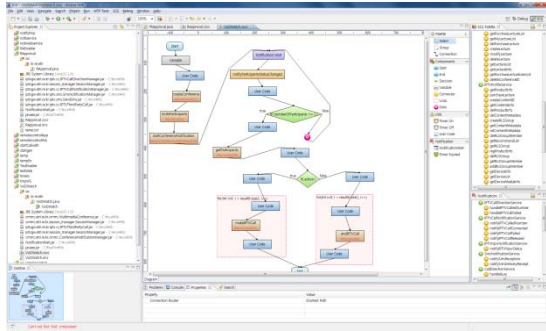
코드 생성기는 서비스 생성도구 사이트의 클라이언트 스텝을 이용하여 Open API 를 호출하는 프로그램 코드를 생성하도록 설계되었다.

본 논문에서 개발한 코드 생성기는 C# 언어로 개발되었으며, standalone 프로그램으로 구동된다. 그래픽 에디터에서 생성한 중간코드를 입력으로 각 컴포넌트 단위별 프로그램 코드 생성과 전체 서비스 로직 흐름을 위한 서비스 컴포넌트, 유틸리티 컴포넌트의 코드 생성을 거쳐 전체 서비스 로직의 프로그램을 생성해 낸다. 이때, 생성된 서비스 로직 프로그램은 java 언어로 출력되어 로컬에 파일 형태로 저장된다.

## 2.3 서비스 디플로이어

저장된 서비스 로직 프로그램은 eclipse 와 같은 java IDE(Integrated Development Environment)를 통해 컴파일하여 실행 가능한 형태로 변환한다. 본 논문에서 개발한 코드 생성기는 두 종류의 서비스를 생성한다. 하나는 웹 서비스의 형태의 융복합 API 이고, 다른 하나는 웹 어플리케이션이다. 웹 서비스 형태로 생성된 서비스 프로그램은 다시 서비스 생성도구에 컴포넌트로 추가할 수 있으며, 이를 활용해 다시 서비스 로직을 구성에 활용될 수 있는 장점이 있다. 웹 어플리케이션은 웹 브라우저를 통해 직접 서비스 로직을 실행할 수 있다.

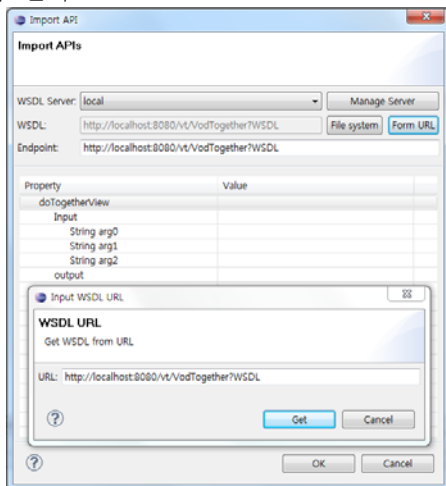
웹 서비스와 웹 어플리케이션 서비스 로직을 컴파일 후 웹 서버에 디플로이하기 위한 서비스 디플로이어는 SailFin 2.0 서버 플랫폼을 기반으로 개발되었다. SailFin 2.0 서버에 웹 서비스 또는 웹 어플리케이션을 디플로이 기능과 연계하여 디플로이에 필요한 설정 파일을 자동으로 생성하여 서버 플랫폼에 디플로이한다.



(그림 3) 캔버스 내 서비스 로직

(그림 3)에서는 그래픽 에디터를 이용하여 생성한 캔버스 내의 서비스 로직을 나타낸다. 이렇게 생성된 서비스 로직은 컴파일 후 서비스 디플로이를 통해 서버 플랫폼에 디플로이된다.

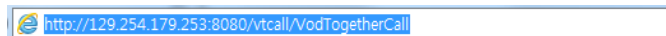
디플로이된 서비스가 웹 서비스일 경우, 다시 서비스 생성도구의 컴포넌트로 추가하기 위해서는 (그림 4)에서 보는 바와 같이 웹 서비스의 WSDL 파일 위치를 지정하여 웹 서비스의 인터페이스를 추출하여 컴포넌트화하게 된다.



(그림 4) 웹 서비스 컴포넌트 추가

디플로이된 서비스가 웹 어플리케이션일 경우에 웹 어플리케이션을 실행하여 실제 서비스 로직의 구동을 확인하고자 할 때는 먼저 웹 브라우저를 구동하여야 한다. 그리고 웹 어플리케이션이 디플로이된 URL 을 입력하여 서비스 로직을 구동하게 된다.

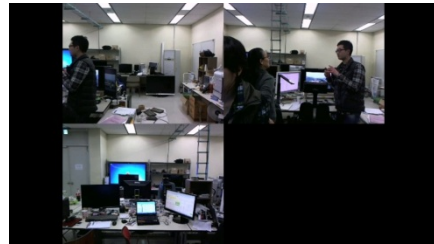
(그림 5)는 웹 브라우저에 서비스 생성도구를 통해 개발되고 디플로이된 웹 어플리케이션의 주소를 입력하게 서비스 로직을 구동하는 방법을 나타낸다.



(그림 5) 웹 어플리케이션 실행

(그림 5)에서 구동한 웹 어플리케이션은 SIP 기반의 영상회의 룬을 개설하고, 각 사용자를 초대하는 기능을 수행하도록 서비스 로직을 구성하였다. (그림 6)은 (그림 5)의 과정을 통해 구동되는 서비스의 실제 실행 화면을 나타내는 것으로 본 논문에서는 개방형 IPTV

시스템과 시험환경을 기반으로 실제 서비스를 구동할 수 있도록 개발하였다.



(그림 6) 웹 어플리케이션 실행

### 3. 결론

본 논문에서는 개방형 IPTV 플랫폼을 기반으로 다양한 IPTV 융합 서비스 개발의 편의성을 도모하기 위한 서비스 생성도구에 대하여 기술하였다. 본 논문에서 소개한 서비스 생성도구는 웹 서비스를 컴포넌트로 하여 그래픽 에디터를 통해 구성한 서비스 로직을 코드 생성기를 통해 웹 서비스 또는 웹 어플리케이션 형태로 생성하고, 서비스 디플로이어를 통해 생성된 서비스를 서버 플랫폼에 디플로이할 수 있는 환경을 제공한다. 현재 SailFin 2.0 서버 플랫폼을 기반으로 개발된 서비스 생성도구는 차후 다양한 RESTful 웹 서비스와 웹 포털에서 제공하는 Open API 들을 수용하여 보다 다양한 융합 IPTV 서비스를 개발할 수 있도록 관련 연구가 진행될 예정이다.

### Acknowledgement

본 연구는 미래부가 지원한 2013 년 정보통신 · 방송 (ICT) 연구개발사업의 연구결과로 수행되었음.

### 참고문헌

- [1] 유초룡, 신영미, 김상기 “개방형 서비스 검증도구의 서비스 액티베이션 방법”, 2006 한국통신학회 학술대회
- [2] 신영미, 유초룡, 정승화, 김상기 “The Creation Method of Open Network Service Application using Service Creation Environment”, ICIN 2006
- [3] 신영미, 유초룡, 정승화, 김상기 “End-user driven Service Creation for Converged Service of Telecom and Internet”, AICT 2008
- [4] O. Droegehorn, I. Konig, M. Belaunde, G. Le-Jeune, J. Cupillard, E. Kovacs, “Professional and End-User-Driven Service Creation in the SPICE platform”, WoWMoM 2008, Newport Beach, USA, June. 2008, pp.1-8.
- [5] R. Lasch, B. Ricks, R. Tonjes, “Service Creation Environment for Business-to-business Services”, WAINA 2009, Bradford, United Kingdom, May. 2009, pp.512-517.
- [6] S. Komorita, M. Ito, and H. Yokota, “Loosely Coupled Service Composition for Deployment of Next Generation Service Overlay Networks”, IEEE Communication Magazine, Jan. 2012, pp.62-72.

- [7] R. Glitho, F. Khendek, A. De Marco, "Creating Value Added Services in Internet Telephony: An overview and a Case Study on a High-Level Service Creation Environment", IEEE Transaction on Systems, VOL.33, November. 2003.