

FavoritesOnTV : 웹서비스 매쉬업을 이용한 스마트 TV 앱 개발

허신영, 박유미, *이태동

한국전자통신연구원, *삼성전자

cyhur@etri.re.kr, parkym@etri.re.kr, [*taedong2.lee@samsung.com](mailto:taedong2.lee@samsung.com)

FavoritesOnTV : Web Service Mashup for Developing Smart TV Applications

Cinyoung Hur, Yoo-mi Park, *Taedong Lee
ETRI, *Samsung Electronics

요 약

본 논문에서는 웹에 공개된 웹서비스들을 매쉬업하여 스마트 TV 앱을 개발하는 과정에서의 문제점을 고찰하고, 문제 해결을 위한 접근방법을 설명한다. 웹서비스 매쉬업 개발 시, 분산된 웹서비스 들 중 개발자가 필요로 하는 적절한 웹서비스를 손쉽게 찾기 어렵고, 웹서비스마다 인증받고 호출하는 방법이 다르며, 또한 서비스의 가용성을 확인하고 이용해야 문제로 요약된다. 이와 같은 웹서비스 매쉬업 과정의 문제를 해결하기 위해 제시한 본 연구의 접근 방법은 검색, 검증, 인증이 손쉽게 지원되는 개발환경 시스템으로 제공될 필요가 있으며, 자동화된 개발환경 시스템은 개발자를 도와 웹서비스의 활용을 증대시킴으로써 다양한 앱 개발의 촉매제가 될 수 있을 것이다.

1. 서론

최근 전자 기기들이 디지털화 하면서 인터넷 기술을 접목하여 눈부신 진화를 거듭하고 있다. 그 중 스마트 TV 는 스크린의 크기와 고화질의 디스플레이 장치 등 외형적 발전뿐 아니라 방송 시청을 비롯하여 통신, 게임, 방송 추천, 정보 검색 등 앱을 실행할 수 있는 스마트 장치로 급속히 진화하고 있다. 이는 방송 시청 위주의 단방향 TV 서비스에서 앱을 통한 사용자와의 소통을 시도하는 양방향 TV 서비스로의 진화를 의미한다. 여기서 TV 앱은 TV 서비스의 진화에 견인차 역할을 하고 있으나, 견고한 앱 생태계를 구축한 스마트 폰 앱과 비교할 때 후발 주자로서 부지런히 그 뒤를 따르고 있는 상황이다. 하지만 스마트 폰에 비해 고화질의 대형 출력 장치, 안정적인 인터넷 연결 환경, 고사양의 프로세서와 저장소, 접근과 사용의 용이성을 가진 스마트 TV 에서 앱의 발전 가능성은 모바일 앱 이상의 잠재력을 지니고 있다고 할 수 있다.

이러한 스마트 TV 앱의 발전은 다양한 입출력 장치의 기능 덕분이기도 하지만 정보의 바다인 인터넷의 자유로운 활용에 기인한다. 인터넷에는 전세계 서비스 업체, 정부, 기관, 학교, 개인들로부터 다양한 정보가 개방되어 있다[1,2]. 특히 RESTful 웹서비스로 개방된 정보는 제 3 자가 HTTP 상에서 손쉽게 정보를 획득하여 앱으로 개발할 수 있다. 스마트 폰에서 애용되는 날씨 앱, 대중교통 운행정보 앱, 지도 앱들이 웹서비스를 통해 인터넷 정보를 활용한 경우이다. 이렇게 웹서비스의 효율성과 접근성에도 불구하고 스마트 TV 에서의 활용은 아직 미흡하다. 개방된 많은 웹서비스들 중에 개발자의 목적에 부합하는 적절한 서비스를 쉽게 찾기 어렵고, 웹서비스가 개방되어 있다고는 하나 웹서비스의 제공자가 그들만의 인터페이스 방식으로 제공하고 있으며, 웹서비스 사용

시 다양하고도 복잡한 인증 절차로 인해 TV 앱에서 적극 활용되지 못하고 있다.

본 논문에서는 7 개 서비스 제공자가 제공하는 7 개의 웹서비스를 매쉬업하여 스마트 TV 앱을 개발하는 과정을 소개하고, 이 과정에서 발생하는 현실적, 기술적, 정책적 문제점을 고찰하고 해결 방법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2 절에서는 웹서비스 매쉬업을 이용한 FavoritesOnTV 앱을 소개하고, 3 절에서는 개발 과정에서의 문제점을 고찰하고 해결 방법을 제시한다. 4 절에서는 FavoritesOnTV 의 실행을 보이며 마지막으로 5 절에서는 본 논문에 대한 결론을 맺는다.

2. FavoritesOnTV 의 설계

FavoritesOnTV 앱은 TV 사용자가 원하는 콘텐츠를 다양한 소스로부터 모아서 한 화면에서 볼 수 있게 하는 앱이다. 즉, TV 의 대형 화면을 충분히 활용하고, TV 입력 장치의 불편을 최소화하기 위해 한번의 검색으로 뉴스, 동영상, 이미지, SNS 멘션을 한 화면에 볼 수 있도록 한다.

그림 1 에서와 같이 FavoritesOnTV 앱은 TV 에서 실행되는 앱으로서, 앱 실행 시 원격에 위치한 서드 파티 웹서비스를 호출하여 콘텐츠를 얻는다. FavoritesOnTV 앱은 화면 틀을 구성하고 서비스 호출 결과가 리스트인 경우 스크롤 기능을 지원하고 (화면 레이아웃 구성 기능), 질의를 통해 데이터를 검색(데이터 검색 기능)하여 선택된 화면 틀에 출력하는 하는 기능(데이터 출력 기능)을 제공한다. 또한 앱 실행 중 리모콘 으로부터 발생하는 이벤트 (음성 입력, 메뉴 선택 등)를 처리하는 기능 (Event Key 처리)을 제공한다.

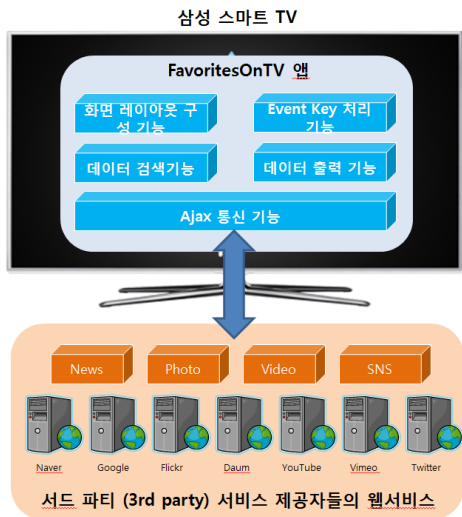


그림 1. FavoritesOnTV 앱의 기능 구조

여기서 FavoritesOnTV 앱은 사용자가 원하는 서비스 제공자를 선택할 수 있도록 다음과 같이 서비스 별 다중 정보를 제공한다.

- 뉴스 검색 웹서비스

사용자가 입력한 검색어로 뉴스를 검색한 결과를 제공하는 서비스로서, 구글(Google)과 네이버(Naver)를 사용하였다.

- 동영상 검색 웹서비스

사용자가 입력한 검색어로 동영상을 검색하여 제공하는 서비스로서, 유튜브(YouTube)를 사용하였다.

- 이미지 검색 웹서비스

사용자가 입력한 검색어로 이미지를 검색하는 서비스로서, 플리커(Flickr)와 다음(Daum)을 이용하였다.

- Social Network Services 검색 웹서비스

사용자가 입력한 검색어로 소셜 네트워크 서비스에서 공개된 메시지를 검색하는 서비스로서 트위터(Twitter)를 이용하였다

3. FavoritesOnTV의 개발과 문제점 고찰

FavoritesOnTV 앱은 삼성 Smart TV SDK 3.5.2[3,4] 버전을 이용하여 개발하였다. 삼성 Smart TV SDK에는 웹앱(자바스크립트, HTML, CSS)과 플래시형 앱 두 종류의 기본 템플릿을 제공하는데 본 연구에서는 다양한 웹서비스를 호출하기 용이한 웹앱을 이용하였다. 본 절에서는 웹서비스를 이용하는 Smart TV 앱을 개발하는 과정에서 개발자들이 겪게

되는 다음의 세 가지 현실적, 기술적, 정책적 문제점을 고찰하고, 문제를 해결하고자 본 연구의 접근 방법을 설명한다.

- 웹서비스 검색

목적에 부합하는 웹서비스를 검색하기 위해서는 기술적으로 UDDI Registry에 접근하면 되지만, 현실적으로는 퍼블릭 UDDI Registry가 활성화되지 않고 있어 일반적으로 ProgrammableWeb [1]이나 Membrane [2]과 같은 포털을 이용하게 된다. 일례로, ProgrammableWeb에는 현재 9,200여개의 API가 등록되어 있고 다양한 메타 정보를 통한 다면 검색(태그, 카테고리, 제공회사, 프로토콜, 데이터 포맷)을 지원하고 있다. 그러나, FavoritesOnTV가 이용하고자 하는 웹서비스를 검색하기 위해 태그에 'news'를 입력하면 총 142개의 웹서비스가, 'image'를 검색하면 108개의 웹서비스 'video'를 입력하면 303개의 웹서비스가 검색된다. 또한, 'sns', 'social network service', 'mention' 등으로 검색하면 웹서비스가 하나도 검색되지 않는 반면 'twitter'로 검색하면 총 107개의 검색결과를 얻는다. 이에 표 1에 정리된 웹서비스를 선정하기 위해, 다양한 검색 조건(태그+제공회사,태그+제공회사+프로토콜, 제공회사+프로토콜)을 조합해가며 원하는 웹서비스를 수동 검색하였다. 여기서 너무 많은 검색 결과는 불필요하거나 관련없는 결과를 필터링하는데 많은 시간을 소요하게 만들고 있었으나 현재까지는 개발자가 스스로 해결해야 하는 문제로 남아있다.

- 웹서비스 호출 방법

웹서비스 검색과 선정을 마친 후, 서비스를 호출하기 위해서는 각 서비스 별로 자바스크립트 파일에 호출부를 정의해야 한다. 특히 RESTful 웹서비스는 SOAP 서비스와 달리 표준화된 통신 규약이 없기에 서비스 제공자 별로 다양하게 인터페이스를 정의하고 있다. 표 1에서 보듯이 웹서비스 별로 호출하는 방법과 응답 형태가 상이한데, 상이한 형태별로 의존적인 프로그램을 한다면 새로운 웹서비스를 이용하고자 할 때 소스 코드의 재활용이 어려울 뿐 아니라 개발의 노하우가 전파되지 못하는 한계가 있다.

이에 본 연구에서는 웹 브라우저에서 호출한다는 공통점을 활용하여 재활용 가능한 템플릿을 정의하였다. 서비스의 주소와 요청 파라미터, 콜백 함수를 입력할 수 있는 템플릿을 이용하여 이종의 웹서비스 호출에 통일성을 제공하였다. 따라서 FavoritesOnTV 앱 개발 과정에서 웹서비스 호출 템플릿을 이용하여 총 7종의 코드 스니펫이 개발되었고(그림 2), 그림 3은 그 중 유튜브 서비스 호출 예를 보여준다.

표 1. FavoritesOnTV에서 이용된 웹서비스

서비스 종류	서비스 URL	서비스 제공자	접근 프로토콜 (호출 방법)	서비스 인증 방법	결과 메시지 형식
뉴스	http://ajax.googleapis.com/ajax/services/search/	Google	REST	key	json
	http://openapi.naver.com/search	Naver	REST	key	XML
동영상	https://gdata.youtube.com/feeds/api/videos	YouTube	REST	key	json
	http://vimeo.com/api/rest/v2	Vimeo	REST	Oauth 1.0	json
이미지	http://api.flickr.com/services/rest/	Flickr	REST	key	json
	http://apis.daum.net/search/vclip	Daum	REST	key	json
SNS	http://search.twitter.com/search.json	Twitter	REST	key	json

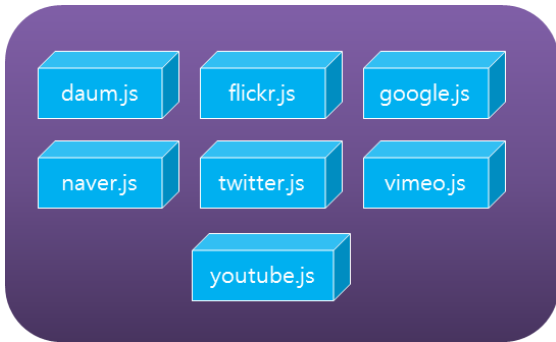


그림 2. FavoritesOnTV 의 웹서비스 호출 코드

```

var youtube = {
  url : 'https://gdata.youtube.com/feeds/api/videos',
  type : 'GET',
  input : {
    'q' : 'smart tv', // 검색어
    'orderby' : 'published',
    'start-index' : 2,
    'max-results' : 10,
    'v' : 2,
    'alt' : 'json' // 메시지 포맷
  },
  ...
};
// 웹서비스 호출
youtube.searchVideo = function(query) {
  youtube.input.q = query;
  $.ajax({
    type : youtube.type,
    url : youtube.qs(youtube.url, youtube.input),
    success : youtube.callback,
    dataType : "jsonp"
  });
};
// 플래시 콘텐츠 재생
youtube.loadVideo = function(url, autoplay) {
  ...
};
// 콜백 함수, 응답 메시지 처리
youtube.callback = function(data) {
  ...
}
    
```

그림 3. 웹서비스 호출 템플릿을 이용한 코드 스니펫

• 웹서비스 보안 제약

TV 플랫폼은 데스크톱이나 모바일에 비해 웹서비스를 활용하는데 보안 제약사항이 더 많다. TV 의 웹 브라우저는 데스크톱과 달리, 크로스 도메인의 통신을 안전하게 해주는 플러그인의(예를 들어, Active X)의 설치가 불가능하다. 그래서 웹 브라우저 자체의 기술만을 활용하여 보안 제약을 우회해야 하는데, 웹서비스 제공자가 이를 지원해야 가능하기에 현실적으로 폭넓은 종류의 웹서비스를 매쉬업하는데 어려움이 있는 실정이다.

본 연구는 서로 다른 도메인에서 제공하는 서비스를 호출하기 위해 JSONP (JSON Padding)를 이용하였다[5]. 이 방법은 매쉬업 서비스를 개발하는데 가장 널리 사용되는 방법으로, JSON 데이터를 자바스크립트 콜백 함수로 감싸고 <script src="..."> </script> 태그 안에 삽입함으로써 웹서비스의 응답을 처리한다. 그러나 이 방법은 악의적인 공격에 취약하므로 웹서비스의 안전성과 보안 수준을 면밀히 검토해야 한다.

FavoritesOnTV 앱은 동영상 검색 서비스로 유튜브 서비스를 이용하였고, 그림 4 와 그림 5 는 유튜브 검색 서비스에 대한 요청 메시지와 응답 메시지이다. 검색어 “ smart tv” 와 JSON 포맷 서비스를 호출하였고, 임의로 생성된 콜백 함수로 감싸진 응답 메시지를 받았다. 이 결과에서 플래시 콘텐츠의 주소를 추출하여 재생하였다.

```

Request URL:
https://gdata.youtube.com/feeds/api/videos?q=smart+tv&orderby=published&start-index=2&max-results=10&v=2&alt=json&callback=jQuery191018371691345237195_1369706189953
Request Method:GET
Status Code:200 OK
    
```

그림 4. 유튜브 검색 서비스 예: 요청 메시지

```

jQuery191018371691345237195_1369706189953 (
....
  "category": [{
    "scheme":
"http://schemas.google.com/g/2005#kind",
    "term":
"http://gdata.youtube.com/schemas/2007#video"
  },
  {
    "scheme":
"http://gdata.youtube.com/schemas/2007/categories.c
at",
    "term": "Entertainment",
    "label": "엔터테인먼트"
  }],
  "title": {
    "$t": "삼성전자 (Samsung electronics) TV CF -
스마트에어컨 (Smart Air conditioner) Q9000 초여름편"
  },
  "content": {
    "type": "application/x-shockwave-flash",
    /* 추출한 플래시 콘텐츠의 주소 */
    "src":
"https://www.youtube.com/v/UBGXTURHQI4?version
=3&f=videos&app=youtube_gdata"
  },
  "link":....
);
    
```

그림 5. 유튜브 검색 서비스 예: 응답 메시지

4. FavoritesOnTV 의 실행

삼성 Smart TV SDK 에 내장된 에뮬레이터를 이용하여 TV 와 유사한 환경에서 앱을 테스트한 후, 개발된 앱을 TV 에 설치하여 실행한다. 이를 위해서는 TV 에서 개발자용 계정을 만들어 등록한 후 개발환경에서 TV 로 다운받을 수 있다[4]. 따라서 개발환경과 TV 가 모두 인터넷(동일 서브넷)에 연결되어 있어야 한다.

그림 6 은 Smart TV 2012 년 모델에서 FavoritesOnTV 앱을 실행한 화면이다. 삼성 Smart TV 는 리모콘, 쿼티 키보드, 스마트 키보드, 음성인식 기능 등 다양한 입력 방법을 지원하는데, 그림 6-a 는 음성으로 검색어를 입력받는 화면이다. 그 결과 그림 6-b 와 같이 뉴스, 동영상, 이미지, SNS 검색을 위한 4 개의 웹서비스를 수행하고, 그 결과를 한 화면의 분할된 영역에 출력하였다. 리모콘의 방향키로 각 영역들 사이의 이동과 영역 내 스크롤이 가능하다. 여기에 새로운 검색어를 입력하면 네 가지 영역에서 다시 웹서비스가 호출되고 그에 따른 결과가 출력된다.



(a) 정보 검색을 위한 키워드 (삼성전자) 입력화면



(b) 뉴스, 동영상, 이미지, SNS 검색 결과 동시 출력 화면

그림 6. FavoritesOnTV 실행 화면

5. 결론

본 논문에서는 구글, 네이버, 다음, 트위터 등의 서비스 제공자가 공개한 총 7 개의 웹서비스를 매쉬업하여 FavoritesOnTV 앱을 개발하는 방법을 소개하고 개발 과정에서의 문제점을 고찰하였다.

웹서비스 매쉬업 개발 시, 웹에 분산된 웹서비스들 중 개발자가 필요로 하는 적절한 웹서비스를 쉽게 찾을 수 없고,

발견한다 해도 서비스 마다 인증받고 호출하는 방법을 찾아 숙지해야 하며, 또한 서비스의 호출 시 크로스 도메인 정책을 우회해야 하는 문제로 요약된다. 즉, 웹서비스 매쉬업 과정은 이중 웹서비스의 다양성과 복잡성이 그대로 계승되어 많은 개발 노하우와 시간을 필요로 한다.

본 논문에서 이와 같은 문제를 해결하고자 접근 방법을 제시하였다. 그러나 이러한 접근 방법은 여전히 개발자의 노하우와 수동 작업에 의존하고 있다. 따라서 다양한 스마트 앱의 탄생에 목표로 많은 웹서비스를 활용하기 위해서는 초보 개발자도 손쉽게 매쉬업할 수 있도록 검색, 검증, 인증이 손쉽게 지원되는 개발환경이 구비되어야 할 것이다. 이에 본 논문에서 제시한 문제점들을 해결하여 스마트 TV 앱 개발 시 웹서비스의 활용도를 높이기 위해 시맨틱 서비스 기술을 도입한 개발환경을 연구 개발 중이다[6].

Acknowledgment

본 연구는 삼성전자 VD 사업부의 삼성-ETRI 협력과제인 ‘스마트 TV 기반 시맨틱 서비스 매쉬업 기술 개발’ 과제로 수행하였습니다.

참고문헌

1. Programmable Web, <http://www.programmableweb.com/>.
2. Membrane, <http://www.membrane-soa.org/>.
3. Samsung Smart TV Developer's Forum, <http://www.samsungdforum.com/Guide/>.
4. Testing Your Application on a TV, <http://www.samsungdforum.com/Guide/art00013/index.html>.
5. Jackson, Collin, and Helen J. Wang. "Subspace: secure cross-domain communication for web mashups." Proceedings of the 16th international conference on World Wide Web. ACM, 2007.
6. <http://www.samsungdforum.com/Devtools/SdkArchive/>.