

# 단말기 간 장면 동기화를 위한 효율적인 장면 업데이트 방법

\*박상현 김현철 최지훈 차지훈

\*한국전자통신연구원 방송통신융합연구부

\*sh\_park@etri.re.kr

## Efficient Scene Update Method for Scene Synchronization among the Terminals

\*Sanghyun Park Hyun-Cheol Kim Ji Hoon Choi Jihun Cha

Electronics and Telecommunications Research Institute

### 요약

최근 다양한 단말기를 대상으로 동일한 미디어 콘텐츠, 애플리케이션 서비스를 제공하는 N-Screen 전략이 두드러지고 있다. N-Screen 서비스를 제공하는 방법의 하나로 단말 간 유사한 사용자 인터페이스를 제공하는 방법이 있다. 이는 사용자에게 친숙한 사용자 경험을 제공하는 장점이 있으며, 이러한 방법을 위해서는 단말 간 장면 동기화가 매우 중요하다. 기존의 N-Screen 서비스에서는 장면 동기화를 위하여, 단말 간 API 인터페이스 등을 사전에 조율해야 하는 문제점이 존재한다. 이에 본 논문은 상기와 같은 N-Screen 서비스의 동기화 방법의 문제점을 해결하기 위하여, 명령어 혹은 이벤트 중심이 아닌 자원 중심의 단말 간 통신 방법에 대해 제안하였다. 이를 위하여 장면 업데이트의 내용을 효율적으로 XML 형태로 기술하고, 전송하는 방법에 대해 연구하였다.

### 1. 서론

최근 다양한 단말기에 동일한 미디어 콘텐츠, 애플리케이션 서비스를 제공하기 위한 N-Screen 전략이 두드러지고 있다. N-Screen 전략은 단말기 간 서비스 연속성을 제공하는 의미에서 크게 주목받고 있다. 그 때문에 애플, 삼성과 같은 미디어 단말기 제조사뿐만 아니라 넷플릭스, 컴캐스트와 같은 미디어 콘텐츠 제공업체에서도 다양한 단말기에 자사의 콘텐츠 서비스 플랫폼을 탑재하기 위한 노력을 기울이고 있다.

현재 N-Screen 전략은 크게 두 가지 유형이 존재한다. 첫째, 단말기 간 동일한 콘텐츠를 공유하는 유형이다. 이 방법은 하나의 콘텐츠를 여러 단말기에서 사용할 수 있는 One-Source Multi-Use의 특징을 가지고 있다. 또 다른 하나는 단말기마다 독립적인 콘텐츠를 제공하는 유형이 있다. 이 방법은 방송 프로그램은 TV를 통해, 웹 콘텐츠는 PC를 통해, 모바일 앱은 모바일 단말기를 통하는 것과 같이, 각 단말기의 특성에 적합한 콘텐츠를 제공하고자 한다. 이 방법의 특징은 단말기 특성에 대한 구분 없이 접근 가능한 통합 서비스 플랫폼을 구축하고자 하는 것이다.

상기의 N-Screen 전략의 유형 중 단말기 간 동일한 콘텐츠를 공유하는 유형에서는, 단말기 간 유사한 사용자 인터페이스와 사용 경험을 제공하는 방법이 주로 사용된다. 이 방법은 콘텐츠 서비스가 제공되는 단말기 간 차이점에 따른 괴리감을 줄이고, 사용자가 서비스에 더욱 쉽게 접근할 수 있도록 돕는다. 상기와 같은 콘텐츠 서비스에서는 동일한 콘텐츠 서비스로 연결된 단말기 사이의 장면이 서로 유사하게 유지되어야 한다. 만약 단말기 중 하나에서 사용자 인터랙션이 일어나 A라는 반응이 발생하였을 때, 다른 단말기에서도 A라는 반응이

일어나야 한다. 이와 같은 동작을 위해서는 단말기 간 장면 동기화가 매우 중요하다. 이와 같은 단말기 간 동기화를 위해 기존에는 이벤트가 발생하였을 때 호출되는 단말기 간 API 인터페이스를 사전에 조율하는 방법을 주로 사용하였다. 그러나 이러한 방법은 단말기 간 서로 이종의 프로그램 언어를 사용하는 경우에는 사용하기 어려우며, API 인터페이스가 미리 정의되지 않으면 시스템 개발에 차질을 빚을 수 있는 단점이 존재한다.

이 논문에서는 상기와 같은 문제점을 해결하기 위해서 계층적 형태로 기술하는 장면 표현 방법과 자원 기반 인터페이스를 제안한다. 인터페이스에 사용하는 자원의 주 형태는 현재 데이터를 주고받는 방법 중 가장 확장성과 범용성이 높은 XML이며, XML은 포함하고 있는 내용을 계층적으로 구성할 수 있는 장점을 가지고 있어, 장면 표현 방법에 직접 적용이 쉽다.

### 2. 장면 표현 기술과 자원 기반 인터페이스

#### 가. LAsER

LAsER(Lightweight Application Scene Representation)<sup>[1]</sup>는 MPEG의 장면 표현 기술 표준의 일종이며, 리치미디어 서비스를 위한 장면 기술 정보를 제공하는 방법을 정의한 규격이다. LAsER는 XML로 기술되며, LAsER 기술 내용은 리치미디어 장면 기술 부분과 미디어 기술 부분으로 구성되어 있다. LAsER의 구성 미디어로는 정지 영상, 동영상, 오디오 및 폰트 등이 있고, 장면 기술 부분은 텍스트, 그래픽스, 애니메이션, 대화형 기능 및 미디어 객체들의 시공간적 구성 정보로 구성되어 있다.

LAsER 장면 기술은 크게 3가지 기능을 제공한다. 첫째, 장면을

구성하는 미디어 객체들의 시공간적 구성 방법과, 둘째, 미디어 객체 간의 상호 작용 방법이다. 마지막으로 장면이 변경될 경우, 변경된 부분을 반영하는 방법을 제공한다.

상기의 기능을 제공하기 위하여 LAsEr는 W3C의 SVG Tiny 1.2에 기반을 두었으며, SVG 규격에서 지원하고 있지 않는 기능들은 LAsEr 확장을 통해 지원하도록 하였다. 또한, 이벤트와 커맨드 기능을 활용함으로써, 장면에 나타난 미디어 객체를 변경/삭제할 수 있게 하고, 새로운 미디어 객체를 추가/삽입할 수 있게 함으로써 보다 효율적으로 상호 작용하는 장면 구성이 가능하게 한다.

## 나. REST

REST(Representational State Transfer)<sup>[2]</sup>는 분산 시스템을 위한 소프트웨어 아키텍처의 한 형식이다. REST는 분산 시스템에서 사용하는 자원을 정의하고, 자원에 대한 주소를 지정하는 방법에 대한 개괄을 의미한다. 자원은 주로 XML 문서가 사용되며, 통신 방식은 HTTP 프로토콜이 사용된다.

기존의 XML, HTTP 프로토콜을 사용하는 SOAP(Simple Object Access Protocol)<sup>[3]</sup>과의 다른 점은 SOAP은 서비스 지향 구조인데 반해, REST는 자원 지향 구조인 점이다. SOAP 기반 웹 서비스는 서비스를 자유자재로 정의할 수 있다는 장점이 있지만, 동시에 복잡하다는 단점을 가지고 있다. 이와 같은 문제점을 해결하기 위해, REST는 자원 기반인 웹과 웹에서 사용하는 HTTP 명령어(GET, PUT, POST, DELETE)를 그대로 사용한다. 이 점은 프로그램을 작성할 때 사용하는 특정 변수에 대한 기본조작 함수인 CRUD(CREATE, READ, UPDATE, DELETE)와 대응되는 장점이 있다. 상기와 같은 특성을 이용하여, 기존 웹 서버와 웹 기술을 이용하여 간단하게 분산 시스템 인터페이스를 구현할 수 있도록 하였다.

## 3. LAsEr와 REST 기술을 이용한 단말기 간 장면 업데이트 방법

본 장에서는 LAsEr를 이용하여 유사한 장면 서비스를 받는 여러 단말기 중 어느 하나의 단말기에 이벤트가 발생하였을 때, 일어나는 반응을 XML 형태의 자원으로 구성하고, 구성된 XML 자원을 다른 단말기로 전달하는 방법에 대해 설명한다.

단말기 간 직접 통신을 하지 않는 대신, REST 기술을 사용하므로 별도의 웹 서버가 필요하며, 단말기와 웹 서버 사이의 통신은 HTTP 프로토콜을 사용하게 된다.

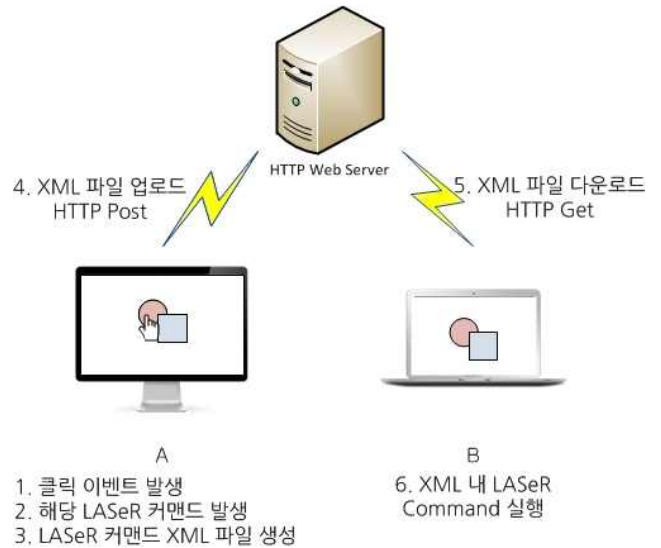


그림 1 제안하는 단말기 간 장면 업데이트 방법

그림 1은 본 논문에서 제안하는 장면 업데이트 방법의 개념도이다. A, B 단말기는 LAsEr를 사용하여 유사한 장면 표현 서비스를 받고 있으며, 서로 동기화되어 있는 상태이다. A 단말기에서 사용자가 이벤트를 발생시켰을 때, 관련된 LAsEr 커맨드가 동작하게 되어 A 단말기의 장면이 변화가 발생한다.

```
<saf:SAFSession xmlns:saf="urn:mpeg:mpeg4:SAF:2005"
xmlns:lsrc="urn:mpeg:mpeg4:LAsEr:2005">
  <saf:sceneUnit>
    <lsrc:Replace ref="circle" attributeName="fill"
value="yellow"/> .....(1)
  </saf:sceneUnit>
  <saf:endofSAFSession/>
</saf:SAFSession>
```

그림 2-(1)은 이벤트가 발생하였을 때, 동작하는 LAsEr 커맨드에 대한 예시이며, 그림 2 전체는 LAsEr 커맨드 전체를 XML 파일 형태로 구성한 것이다. 이처럼 LAsEr 커맨드는 계층적 구조를 따르고 있으므로 쉽게 XML 파일 형태로 구성할 수 있다.

상기와 같이 구성된 LAsEr 커맨드 XML 파일은 HTTP Post 명령을 사용하여, 웹 서버에 업로드 된다. B 단말기에서는 HTTP Get 명령을 사용하여 업로드 된 파일을 다운로드한다. B 단말기에서는 직접 이벤트가 발생하지 않아도 다운로드한 LAsEr 커맨드를 실행하여, A 단말기와 유사한 장면 변화가 일어나도록 하여, 동기화를 유지할 수 있다.

## 4. 결론

본 논문에서는 LAsEr 장면 표현 기술과 REST 웹 인터페이스 기술을 사용하여, 단말기 간 장면 동기화 방법을 제안하였다. 이와 같은 방법은 단말기 간 유사한 사용자 인터페이스를 사용하는 서비스 시나리오에서 간단하게 장면 동기화를 구성하는 방법으로 사용될 수 있다. 제안된 방법은 단말기 간 구현된 프로그래밍 언어의 차이, 네트워크

종류의 차이 등을 고려할 필요가 없기에, 동기화 구성이 더욱 간편해진다. 이는 장면 동기화 서비스 시나리오에 필요한 기술적 진입 장벽을 낮춰, 서비스를 더욱 활성화할 수 있는 장점이 있다.

### 참고 문헌

- [1] ISO/IEC 14496-20: Information technology - Coding of audio-visual objects - Part 20: Lightweight Application Scene Representation (LAsE) and Simple Aggregation Format (SAF)
- [2] Roy T. Fielding, Richard N. Taylor, "Principled Design of the Modern Web Architecture", ACM Transactions on Internet Technology, Volume 2 Issue 2, May 2002
- [3] Thomas Erl, Service-Oriented Architecture (SOA), Prentice Hall, August, 2005

### Acknowledgement

본 연구는 방송통신위원회의 방송통신미디어 원천기술개발사업의 일환인 "IPTV용 Interactive 시점제어 기술개발[09912-03002]" 과제  
의 결과물임을 밝힙니다.