정규논문 (Regular Paper)

방송공학회논문지 제18권 제1호, 2013년 1월 (JBE Vol. 18, No. 1, January 2013)

<http://dx.doi.org/10.5909/JBE.2013.18.1.43>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

DOM update를 이용한 효율적인 멀티미디어 장면 구성 정보 변경 방안

김 규 현^{a)†}, 박 정 욱^{a)}, 김 병 철^{b)}

Effective Method to Change Multimedia Scene Configuration Information Using DOM Update

Kim Kyuheon^{a)†}, Park JungWook^{a)}, and Kim Byungchul^{b)}

요 약

리치미디어 서비스는 비디오, 오디오, 텍스트와 같은 다수의 멀티미디어 요소들을 동시에 소비할 수 있으며 시청자의 요구사항을 수용하여 개인화된 데이터에 대한 이용이 가능한 대화형 미디어 서비스를 뜻한다. MPEG(Moving Picture Expert Group)에서는 BIFS(Binary Format for Scenes)와 LASER(Light Application Scene Representation)를 제정함으로써 리치미디어 서비스를 위한 방안을 제시하였다. 상기 두 표준은 다수의 미디어 콘텐츠간의 시간적, 공간적 배치에 대한 정보와 콘텐츠 간 상호작용을 기술한 장면 구성 정보를 이용함으로써 리치미디어 서비스를 가능하게 하였다. 그러나 기존의 리치미디어 서비스에서는 사용자 및 서비스에 의해서 장면 구성 정보가 변경될 때마다 매번 단말이 가진 기존의 장면 구성 정보를 삭제하고 새로운 장면 구성 정보를 생성 및 분석해야 하는 비효율적인 방법으로 장면 전환이 이루어진다. 따라서 본 논문에서는 DOM(Document Object Model) update라는 웹문서 데이터 인터페이스 규격을 사용하여 보다 효율적이고 동적인 장면 구성 변경 방안을 제시하여 장면 전환의 시간적 단축 효과와 장면전환에 따른 화면의 정지 등의 불연속성을 제거하는 효과를 검증하였다.

Abstract

Richmedia Service means that interactive media service can provide view with various multimedia elements(such as Video, Audio, Text) at same time. Various Multimedia elements can be serviced by Scene Description technology standards like BIFS(Binary Format for Scenes) and LASER(Light Application Scene Representation). By providing Scene Component information, richmedia service is available to various multimedia services. so users is available to personalized services fitting temporal and spatial options. In conventional technology, when the scene is changed by user or service, mobile deletes the scene of configuration information and makes new scene of configuration information. this is a very inefficient way. In this paper, Propoesed that by using DOM(Document Object Model) method, to pass only the dynamic configuration part, changes scene method.

Keyword : BIFS, LASER, Scene Description, DOM script

a) 경희대학교 전자전파공학부(College of Electronics and Information, Kyung Hee University)

† Corresponding Author : 김규현 (Kim Kyuheon)

E-mail: kyuheonkim@khu.ac.kr

Tel: +82-2-3408-3753 Fax: +82-2-3408-4321

※ 이 논문은 지식경제부의 지식경제 기술혁신사업("스케일러블 데이터기반 Edutainment 핵심기술 개발")의 연구결과로 수행되었음(과제번호: 10035486).

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학IT연구센터육성 지원사업의 연구결과로 수행되었음" (NIPA-2012- H0301-12-1006)

· Manuscript received September 13, 2012 Revised November 29, 2012 Accepted December 27, 2012

I. 서론

다양한 미디어 서비스와 단말기의 발전으로 시청자는 비디오, 오디오, 텍스트, 애니메이션과 같은 다수의 멀티미디어 콘텐츠들을 동시에 소비할 수 있게 되었다. 3D 콘텐츠, N-Screen, 멀티뷰 서비스와 같이 다수의 콘텐츠를 동시에 소비하고자 하는 소비자의 성향이 점차 증가됨에 따라 미디어 서비스 제공자 입장에서는 각 미디어들 간의 소비시간과 공간정보를 제어할 수 있는 기술이 필요하게 되었다.

이러한 기술적 요구사항이 증가되면서 MPEG(Moving Picture Expert Group)에서는 각 미디어 객체들을 통제하고 스트림 지원이 가능한 서비스에 대한 방안이 관심 가지기 시작했다. 그래서 MPEG은 사용자가 직접 원하는 콘텐츠에 대한 정보 교환과 시공간적으로 제어할 수 있는 대화형 서비스의 지원을 위해 장면 구성 정보를 활용한 BIFS(Binary Format for Scenes)^[1]와 LAsER(Light Application Scene Representation)^[2]라는 표준 규격 제정하였다.

여기서 장면 구성 정보는 장면을 구성하는 멀티미디어 요소들을 객체로 표현함으로써 각 객체가 화면에서 차지하고 있는 영역에 대한 공간적 정보와 어느 시점에 소비가 이루어져야 하는지에 대한 시간적 정보를 지정하여 객체들 간의 연관관계를 나타내는 정보를 가진다. 뿐만 아니라 시청자의 요청에 대한 즉각적인 피드백이 가능하도록 이벤트 기능을 지원하며, 다양한 단말에서 미디어 객체 속성정보에 대한 접근이 용이하도록 장면 구성 정보를 이진화 정보 또는 XML(eXtensible Markup Language)^[3]언어 형태로 작성되었다.

BIFS 또는 LAsER 표준에 따르는 단말이 장면 구성 정보를 분석하게 되면 단말은 메모리상에서 DOM(Document Object Model) tree가 생성한다. DOM Tree는 HTML(Hypertext Markup Language) 또는 XML 문서 등을 요소별로 편집 가능하도록 플랫폼/언어 중립적인 인터페이스로써 웹 페이지를 보다 인터랙티브하게 구성할 수 있다. 이렇게 DOM Tree를 사용하는 단말에서 장면의 변경되어야 하는 상황이 발생하였을 때 새로운 장면에 대한 DOM Tree의 편집이 이루어져야 한다.

이때 DOM update라는 방법을 사용할 시 단말이 매번 장

면 구성정보를 분석하지 않고도 메모리상에서 부분적 또는 전체적으로 변경이 가능함으로써 단말에 대한 부담이 적어지게 된다.

HTML5와 같은 W3C 표준에서는 DOM update를 사용함으로써 메모리상 객체들에 대한 자유로운 변경에 관한 방안이 표준으로 제정되어 있으나 MPEG 표준에서는 BIFS 및 LAsER와 같은 장면 구성 정보를 이용하는 표준에서 DOM update를 적용 가능함에도 불구하고 아직 변경에 대한 방안이 언급되지 않아 장면이 변경될 때마다 매번 새로운 장면 구성 정보에 대한 DOM tree를 생성해주는 장면 변경에 번거로움이 발생하게 된다. 이러한 방식은 단순히 텍스트 및 이미지 요소로 구성되어 있는 웹페이지 정보만 포함된 XML 문서의 경우 재구성에 대한 단말의 부담이 되지 않지만 비디오, 오디오, App과 같은 다양한 멀티미디어 요소들로 구성된 장면 구성 정보의 경우 환경이 제한된 단말에서는 큰 부담적인 요인으로 작용될 수 있다.

본 논문은 상기에서 언급한 장면이 변경될시 발생하는 단말의 부담을 줄이기 위해서 BIFS와 LAsER와 같은 MPEG 표준에서도 DOM update를 이용하여 효과적인 장면 구성 정보의 변경 방안을 제시하고자 한다. 이는 장면의 변경되어야 하는 요소만 부분적으로 편집이 됨으로써 보다 효율적인 멀티미디어 장면 구성 기술 서비스가 이루어질 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 장면 구성 정보를 이용한 리치미디어 표준 기술인 BIFS와 LAsER의 소개와 메모리 상 객체에 대한 논리적 구성을 이루는 DOM script, 그리고 DOM을 활용하는 HTML5 예제를 통한 기존 기술에 대한 설명과 3장에서는 논문에서 장면 구성 정보 변경 방안을 제안한다. 4장에서는 실험 결과에 대한 성능을 검증하고 5장에서 본 논문의 결론을 맺는다.

II. 장면 구성 기술

IT 단말기의 발전과 미디어 서비스의 진보는 시청자가 원하는 콘텐츠와 정보를 이용할 수 있는 차세대 멀티미디어 서비스 시장을 형성하였다. 이와 같은 시장 인프라의 구

축에 선도한 대표적인 표준 기술로는 다중화 된 미디어 콘텐츠들을 융합 및 장면 구성 정보로 처리할 수 있는 BIFS 및 LASCeR가 있다. 상기 두 표준은 비디오, 오디오, 정지영상 및 2D/3D 그래픽과 같은 미디어 콘텐츠들에 대한 장면 구성 기술을 계층적인 형태로 기술함으로써 독립된 객체간의 상호관계를 정립한다. 또한 객체 간의 시간적, 공간적인 배치와 사용자의 입력이나 시간에 따른 객체의 동작의 접근 및 조작이 가능한 기술을 제공한다.

이에 시청자는 TV, 핸드폰, 태블릿PC, PC 등의 다양한 단말 환경에서 대화형 서비스와 효율적인 데이터 처리가 가능하며 현재 BIFS 표준을 활용하여 우리나라에서 세계 최초로 상용화한 지상파 DMB(Digital Multimedia Broadcasting)의 데이터 서비스로 상용화되었으며 LASCeR의 경우 무선포털, 휴대형 TV 등을 이용한 다양한 형태의 서비스의 적용이 가능하다.

2장에서는 장면 구성 기술을 이용한 리치미디어 서비스 표준 기술인 BIFS와 LASCeR를 소개하고, 현재 장면 변경에 사용되는 DOM method와 이를 활용하는 HTML5 예제를 들고자 한다.

1. BIFS(Binary Format for Scene)

MPEG에서 제정한 MPEG-4 시스템은 콘텐츠에 대한 대화형 서비스의 이용이 가능하며 서비스 이용 측면에서 시청자가 원하는 멀티미디어 요소에 대한 접근, 추가, 삭제 등이 가능하다. 또한 MPEG-4는 장면을 구성하는 각각의 콘텐츠들을 요소로 저장하여 추후 장면에서 동일한 콘텐츠가 이용될 시 재사용이 가능하여 저장측면에서도 효율적인 장점이 있다.

이러한 MPEG-4 시스템의 기능적 이점은 장면 구성 정보를 제공하여 장면을 구성하는 다수의 미디어 객체들을 분류하고 이에 대한 시공간적 정보와 상호간의 연관성, 사용자의 요청에 따른 이벤트 정보를 처리함으로써 가능하다. 여기서 장면이란 단말의 화면에서 표현할 수 있는 비디오, 오디오, 이미지, 텍스트 등과 같은 멀티미디어들의 구조적 형태를 뜻하며 이는 장면 구성 정보를 통해 각 객체들 간의 시공간적 소비 방안을 제시한다.

BIFS는 대표적인 MPEG-4 시스템의 장면 구성 정보의 하나로써 장면을 구성하는 각 미디어 콘텐츠 요소들을 텍스트 문서 형태로 기술하는 언어인 VRML(Virtual Reality Modeling Language)을 기반으로 두고 있다. VRML은 2차원과 3차원으로 표현되는 가상현실을 구조화하는 언어로써 화면 내부의 객체들을 계층적 형태로 표현이 가능하다. 그러나 VRML에서 사용하는 신택스(Syntax)와 장면의 기술 방식은 장면 업데이트 명령에 대한 장면 서술 정보가 없어 장면 전환에 있어 어려움이 발생한다. 이에 실시간 2D 비디오/3D 비디오/오디오/이미지에 대한 객체 정보와 장면의 전환에 있어 이진 부호화 형태의 BIFS 표준 기술을 통해 장면 전송 및 변경에 대한 효율적인 방안을 제시한다.

BIFS는 초기 객체 서술자(Initial Object Descriptor), 장면 서술자(Scene Descriptor), 객체 서술자(Object Descriptor)들로 구성되어 있다. 초기 객체 서술자는 가장 먼저 복호화가 이루어지는 장면 구성 정보로 장면을 구성하는 각 미디어들 간의 프로파일(Profile)과 레벨(level)을 설명하고 BIFS 스트림과 객체 서술자 스트림(Object Descriptor Stream)을 연결하기 위한 기초 스트림 서술자(Element Stream Descriptor)를 포함한다^[5]. 기초 스트림 서술자는 미디어 데이터 또는 장면 기술 정보의 압축된 데이터를 포함하고 있으며 포함된 데이터 별로 서로 다른 디코더에 입력되어 복호화가 이루어진다. 장면 서술자는 콘텐츠를 구성하는 각 미디어 객체들이 소비되는 시공간적 정보와 객체간의 계층적인 구조를 기술한 정보를 뜻한다. 이 정보를 이용하여 단말기는 미디어 객체들 간의 상호소통과 재사용이 가능하다. 객체 서술자는 장면 구조를 구성하는 객체들을 노드(Node)라 호칭하면서 시청각 스트림에서 필요로 하는 노드와 이들을 실제 멀티미디어 데이터를 연결하는 중간자 역할을 한다.

그림 1은 MPEG-4 시스템 단말기에서 장면 구성 정보를 사용하여 미디어 콘텐츠를 재생하는 과정을 나타낸다. 단말기는 먼저 초기 객체 서술자(Initial Object Descriptor)를 수신 받아 장면 서술자(Scene Descriptor)와 객체 서술자(Object Descriptor)에 대한 초기 정보를 분석한다. 객체 서술자를 통해 각 미디어의 프로파일과 레벨에 대한 정보를 파악하고 장면 서술자 정보로 각 장면을 구성하고 있는 미

그림 1. BIFS를 이용한 장면 구성 방법
Fig. 1. Scene Composition method using BIFS

디어 콘텐츠들 간의 계층별 논리적 구조를 형성한다.

장면 서술자는 단말에 장면을 구성하는 미디어 객체의 요소와 속성 정보를 제공하며 각 객체들에 대한 ID를 부여함으로써 특정 객체에 대한 접근, 조작, 편집이 가능하게 도와준다. 특정 시간이나 시청자의 요청에 따른 장면이 전환이 이루어질 경우, BIFS Command를 통해 특정 객체에 대한 독립성을 보장되는 장면 구성 변경이 가능하다.

즉 단말은 BIFS 기술을 통해 장면을 구성하는 멀티미디어 장면 그래프를 형성하여 각 객체들 간의 시간적/공간적 정보를 객체에게 지시하여 각 객체들 간의 동기화 및 미디어에 대한 제어방법 그리고 객체들 간의 동적인 장면 구성을 위한 정보를 제공할 수 있다^[6]. 또한 BIFS 데이터들은 Binary 형태로 사용됨으로써, 압축의 효율성을 높이면서 사용자가 요청한 피드백에 대한 미디어 정보를 제공할 수 있는 대화형 서비스 지원이 가능하다.

2. LAsER(Light Application Scene Representation)

LAsER는 MPEG-4 시스템 규격에서 장면 구성 정보를 활용하는 리치미디어 서비스 지원 가능한 표준 기술의 또 하나의 예이다. 이는 풍부한 장면 기술 지원과 높은 압축률을 지닌 시스템적 특징을 가진 표준으로 다양한 단말환경에서 BIFS보다 효율적인 기능 지원이 가능하다^[5]. 휴대폰, 태블릿 PC와 같은 단말기와 제한된 네트워크 상황에서 MPEG-4의 정의한 동적인 상호작용에 대한 정보처리가 이루어지기 위해선 무엇보다 데이터에 대한 연산효율이 중요하다. LAsER는 다양한 프로토콜 상에서 전송이 가능하며 효율적인 미디어 데이터 복호화가 이루어질 수 있도록 그림 2와 같은 형태로 구성되어 있다.

LAsER의 장면을 구성하는 각 media element들은 2D 그래픽 표현에 이용되는 SVG(Scaleable Vector Graphics) tiny 1.2규격으로 표현된다. 이는 미디어 장면을 XML 문서 형태로 기술되어 XML 문서 분석과 SVG 지원이 가능한 휴대폰, 태블릿 PC, 노트북과 같은 제한된 단말환경에서도 다양한 벡터 그래픽 효과 및 멀티미디어 기능을 구현할 수 있다^[6]. 또한 미디어 콘텐츠들을 독립적인 객체로 분류함으로써 각 객체 간의 논리적 구조를 Scene Tree형태로 메모리에 적재된다.

LAsER Extentions는 SVG에서 지원하지 않는 Progres-

sive Rendering mode와 같은 서비스 방식 과 전송 규격에 적합하게 하나의 장면을 다수의 전송 패킷으로 분리할 수 있다. 또한 장면을 구성하는 객체들을 Scene Tree 구조로 형성하여 각 객체들에 대한 직접적으로 지시함으로써 삽입, 편집, 제거와 같은 명령을 통해 장면 전환을 수행 할 수 있다. 이러한 SVG Scene Tree와 LAsER Extensions를 통틀어 LAsER Commands라 통칭하며 대표적으로 사용되는 Command의 종류와 기능은 다음과 같다.

- NewScene : 새로운 장면 생성
- RefreshScene : 현재 장면을 새로 구성하여 Packet의 손실에 대한 복구 및 LAsER stream상의 Random Access point에 사용
- Insert : 그룹 안에 특정한 element를 삽입
- Delete : 그룹 안에 특정한 element를 삭제
- Replace : 그룹 안에 특정한 element를 변경
- Add : Replace와 같이 특정 엘리먼트의 속성값을 바꾸는 역할을 하지만 교체 대신 원래의 값에 더하는 것을 의미함
- Save : 리스트 안의 속성 값을 저장
- SendEvent : 장면 안의 특정 요소에 대한 이벤트 전송
- Activate : Scene tree 안의 대기 중인 요소 활성화

그림 3은 실제적인 LAsER Command를 이용한 장면 구

그림 2. LAsER 구조
Fig. 2. Structure of LAsER

그림 3. LAsER를 이용한 장면 구성 변경 예

Fig. 3. Example of Scene Composition using LAsER

성의 변경과정을 나타내며, 이는 SVG Scene Tree 정보를 XML 문서 형태로 구성하여 사용자에게 각 장면마다 여러 미디어들의 소비 방안을 제시한다. 이 소비 방안에 대한 해당 콘텐츠의 구성에 대한 분석이 이루어지기 위해선 XML Parser가 요구하게 된다. 하지만 제한된 단말 환경 안에서 XML 문서 복호화가 여러 번 이루어 질 경우에는 많은 연산량을 요구하게 되며 스트림 서비스가 불가능하다는 문제점을 내재하고 있어 LAsER에서는 독자적인 Binary 방법 및 업데이트 방법을 정의하였다^[6]. 단말은 LAsER를 이용함으로써 제한된 대역폭 및 단말 환경에서도 장면 구성 정보 이용 및 편집에 대한 LAsER Command 처리 효율성을 높이며 시청자의 요청 또는 서비스 제공자에 의한 장면 전환이 가능해졌다. 이때 LAsER Command를 이용하여 SVG Scene Tree를 변경할 시 Tree 내부의 편집 대상을 ID를 통해 식별하여 편집 내용과 변경 시간에 대한 정보를 전달하게 된다. LAsER Command를 받은 객체는 변경 시간에 맞춰 속성 값을 변경시킴으로써 장면의 전환이 이루어지게

된다.

이처럼 미디어 객체들은 각자의 시간 정보들을 가지면서 다른 미디어 객체들과 동기화를 이룰 수 있으며 독립적인 객체들 간의 관리와 재사용을 통한 데이터 압축에 대한 효율성을 높일 수 있다.

3. DOM(Document Object Model) Tree

HTML 또는 XML 문서는 스프레드 시트, 주소록 또는 다른 구조 데이터를 텍스트 형태의 논리적 구조로 나타냄으로써 사람이 읽을 수 있다. 하지만 정작 컴퓨터는 해당 문서의 텍스트 정보만으로 구조적인 내용에 대해 인식이 불가능하다.

이에 W3C(World Wide Web Consortium)에서는 DOM이라는 프로그래밍 인터페이스 문서를 권고함으로써 HTML, XML 문서의 요소들을 독립된 객체들로 구분하여 구조적인 객체 표현 양식과 접근 방식을 정의한 API 규격

을 제공하였다. 이는 DOM을 이용함으로써 문서내의 모든 데이터를 노드의 부모 자식 계층 형태로 메모리에 적재가 가능해졌다.

그림 4(a)와 같은 형태의 XML 문서가 파싱되면 단말기의 메모리상에 그림 4(b)와 같은 DOM tree 구조를 형성된다. 이러한 Tree 구조를 구성하는 객체들을 노드라 호칭하며 각 노드들은 고유 id를 지니고 있어 메모리 상에서 쉽게 접근 및 편집이 가능하다. DOM tree를 사용하는 노드는 엘리먼트 노드(element node), 속성 노드(attribute node), 텍스트 노드(text node)로 분류할 수 있다.

그림 4(b)의 타원 상자는 엘리먼트 노드(Element Node)를 뜻하며 정보 컨테이너를 나타낸다. 이는 XML을 만드는 기본 노드로써 다른 엘리먼트 노드나 텍스트 노드를 포함하며 유일하게 속성을 가질 수 있는 노드를 뜻한다.

그림 4(b)의 평행사변형 상자는 속성 노드를 뜻하며 해

당 엘리먼트 노드에 대한 정보를 포함하고 있다. 속성 노드는 다수의 속성을 포함함으로써 생성된 오브젝트의 property에 접근하여 각 요소에 대한 제어가 가능하게 된다.

그림 4(b)의 사각형 상자는 텍스트 노드를 뜻하며 의미가 있는 텍스트 정보를 포함하거나 혹은 공란이다⁹⁾. 또한, 이러한 DOM 객체와 노드 및 그 속성들은 W3C DOM 표준에 따라 접근 및 조작이 가능하여 createElement(), createTextNode() 같은 노드 생성 및 접근, nodeName, nodeType, tagName과 같은 노드 파악, parentNode(), firstChild() 같은 트리 구조 파악 등의 메소드들이 사용되고 있다.

다음은 W3C DOM 표준에서 제공하는 대표적인 DOM method에 대한 설명을 나타낸다.

- createElement : 엘리먼트 노드를 생성
- createTextNode : 텍스트 노드를 생성

그림 4. DOM tree 예제
Fig. 4. Example of DOM tree

- appendchild : 엘리먼트 노드에 추가
- insertBefore : 이전 노드에 새로운 노드를 삽입
- removeChild : 대상 노드를 제거
- setAttribute : 속성 값을 변경
- getAttribute : 속성 값을 반환
- getElementById : ID에 해당하는 element 검색
- getElementsByTagName : Tag에 해당하는 element를 찾아 배열로 반환

BIFS나 LAsER에서 사용되는 장면 구성 정보들은 XML

기반으로 정의되어 있어 해당 콘텐츠를 재생하려면 XML 텍스트 프로세싱을 위한 Parser가 필요하다. 그러나 XML Parser를 통해 분석된 XML데이터들은 단말기의 메모리 상에 DOM tree를 생성하여 논리적 구조를 형성하며 이러한 정보는 Renderer를 거쳐 사용자에게 구성된 장면을 출력시킨다.

하지만 상기 두 표준기술들은 오직 하나의 장면만을 표현할 수 있으며 서비스 제공자가 지정한 이벤트 또는 사용자의 요청에 의한 장면 변경이 이루어질 때마다 매번 새로운 장면 구성 정보를 생성해야하는 단점이 있다. 그림 5-(a)에서

그림 5. 기존 장면 구성 정보 변경 방법

Fig. 5. Existing methods of Scene Composition information

나타낸 바와 같이 서비스 제공자는 콘텐츠 id가 v1인 Video 객체와 콘텐츠 id가 a1인 Audio객체로 구성된 초기 장면 구성 정보를 XML 형태의 데이터로 생성하여 소비자에게 전달하게 된다. 소비자는 수신 받은 장면 구성 정보를 분석하여 메모리상에 DOM tree 형태로 생성하고 각 요소에 대한 구조화와 복호화를 수행하면서 콘텐츠가 소비된다⁹⁾.

하지만 생성자의 이벤트 또는 소비자의 요청에 대한 장면 구성 정보가 변경될시 그림 5-(b)에서 보듯이 기존 장면에 콘텐츠에 이미지 객체를 삽입해야 할 경우 추가 객체뿐만 아니라 기존 장면 정보까지 중복된 XML문서를 생성해야한다. 소비자 또한 장면을 전환하기 위해 기존의 v1과 a1를 다시 분석한 뒤 id가 i1인 이미지 객체를 삽입해야하는 번거로움이 발생한다.

4. HTML5

HTML5는 웹 개발자들이 실질적으로 부딪히는 문제를 해결하고자 W3C에서 만들고 있는 차세대 웹 표준을 뜻한다. HTML5에서는 HTML4와 호환성을 유지한 채 비디오와 오디오와 같은 미디어 태그를 별도로 정의하면서 플러그인 없이 다양한 멀티미디어 서비스를 제공할 수 있다. HTML5의 표준화로 단순히 요소의 구조적 형태만 제시하던 웹문서는 이제 직접적으로 멀티미디어 서비스가 지원하게 되면서 사용자의 요청에 대한 직접적인 처리 방안이 필요하게 되었다. 이에 W3C에서는 HTML5부터 DOM method 방안에 대한 규격이 제정되었다.

HTML5에서 DOM method를 사용하는 한편의 예로 그

그림 6. HTML5에서 DOM method 사용 예제
Fig. 6. Example of HTML5 using DOM method

림 6에서 나타난다^[8]. HTML5의 요소들은 고유의 메소드, 속성, 이벤트 값을 가지고 있다. 사용자가 웹페이지를 통한 비디오나 오디오와 같은 멀티미디어 요소를 사용할 시 재생, 정지, 크기 변경과 같은 편집을 하기 위해서는 각 요소에 대한 메소드, 속성 값에 대한 변경을 요청해야 한다. 과거의 HTML4에서는 플러그인 없이 요소 값을 변경하기 위해서는 웹페이지를 새롭게 파싱이 이루어졌었지만 HTML5 부터는 DOM을 통해 즉각적인 요소 값에 대한 편집이 가능

하다. 하지만 HTML5에서 DOM 레벨에서 이러한 동적인 장면 변경이 가능하나 이는 웹문서 상에서만 적용된다는 한계를 가지고 있다.

III. 제안 방안

앞서 설명한 바와 같이 BIFS와 LAsER에서는 장면 변경

그림 7. DOM update를 이용한 장면 구성 정보 갱신 방법
Fig. 7. Method of Scene update information using DOM update

이 이루어질 때 매번 장면 구성 정보를 새로 생성하게 되며 이에 대한 분석이 이루어진다. 이 때 장면 변경에 불필요한 요소 데이터도 처리하게 됨으로써 비효율적인 시간적 소비와 연산 과정이 발생하게 된다. 한편 HTML5에서는 변경하고자 하는 요소에 대해 DOM method를 이용한 데이터 생성과 분석을 거쳐 불필요한 시간적 소비와 연산과정 없이 동적인 장면 구성 변경이 가능하다. 하지만 이는 웹페이지 상에서만 적용할 수 있는 한계를 지니고 있다.

따라서 본 논문에서는 BIFS와 LAsEeR 표준에서도 웹페이지 동적 변경 기술인 DOM method를 이용한 새로운 장면 구성 정보 변경 방안을 아래 그림 7과 그림 8과 같이 제안한다.

먼저 서비스 생성자는 그림 7-(a)의 비디오와 오디오 정보가 담긴 초기 장면 구성 정보를 XML 형태로 생성하여 소비자에게 전달한다. 소비자는 먼저 초기 장면 구성 정보를 수신 받아 XML File Parser를 통해 이를 분석하고 메모리상에 DOM Tree를 생성하여 초기 멀티미디어 장면을 만들어 소비한다. 멀티미디어 서비스를 이용하는 도중 서비

스 생성자의 이벤트 또는 소비자의 요청으로 인한 장면이 변경이 이루어져야 할 경우 DOM update를 사용하여 보다 효율적인 장면 구성 정보가 변경이 가능하다.

서비스 생성자는 그림 7-(b)와 같은 추가 이미지 정보가 담긴 DOM update message를 생성하여 이를 소비자에게 전달하게 된다. 소비자는 기존의 메모리 상의 DOM tree는 보존한 채 DOM update message를 XML File Parser로 분석하여 특정 요소에 대한 부분적인 업데이트를 하여 DOM tree를 변경한다. 이러한 방안을 적용시킴으로써 BIFS와 LAsEeR를 이용하는 단말에서도 장면 구성정보를 편리하게 편집이 가능하다. 또한 기존의 멀티미디어 서비스를 보다 다양한 단말에서 단말의 특성에 맞게 적용시킬 수 있을 뿐만 아니라 IPTV, 홈네트워크 등 방송과 통신이 융합된 환경에서 멀티미디어 서비스 제공자와 사용자 사이의 복잡성을 낮추고 효율성을 높일 수 있는 효과를 얻을 수 있다.

그림 8은 DOM update를 이용하여 Video와 Audio로 구성된 초기 장면에 텍스트를 추가하는 과정을 도식화한 것이다.

그림 8. DOM update를 이용한 장면 구성 변경 도식화
Fig. 8. Figure of Scene composition change using DOM update

먼저 복호화가 이루어진 Video와 Audio 데이터는 Scene Generator에 입력하게 된다. Scene Generator는 입력된 정보를 기반으로 하여 각각 미디어들이 언제 어떤 방식으로 소비될지에 대한 정보를 장면 구성 정보에 XML문서 또는 바이너리 정보 형태로 생성한다. 각 Video와 Audio의 동기화에 대한 정보를 포함한 장면 구성 정보는 Packetizer에 입력되어 패킷화되며, 다시 미디어 정보와 함께 Multiplexer를 거쳐 다중화 되어 클라이언트에 전달된다(그림 8-①).

클라이언트의 단말은 RF 또는 IP와 같은 전송망을 이용하여 패킷형태로 미디어 정보를 전송받아 Demultiplexer에 입력한다. Demultiplexer는 패킷을 분석하여 Video와 Audio 정보를 가진 패킷은 Media Decoder에 전송하여 복호화를 이루며, 장면 구성 정보는 Scene Parser에 전송한다(그림 8-②). Scene Parser는 클라이언트의 단말의 메모리에 DOM tree를 생성하여 Renderer에 전달되며 DOM tree에 따라 각 Video와 Audio는 재생 위치 및 시점에 맞춰 미디어 소비가 이루어지게 된다.

이렇게 미디어 서비스 이용 도중 Text요소를 추가해야 할 경우 DOM generator를 이용하여 Text에 대한 DOM update 정보를 클라이언트에게 전달한다(그림 8-③). 그리고 클라이언트는 전달받은 DOM update를 이용하여 메모리에 존재하는 DOM tree에 Text요소를 추가하여 Renderer에 전달함으로써 화면에 텍스트 문구를 추가시킬 수 있다(그림 8-④).

이처럼 DOM update를 이용할 경우 기존 요소들에 대한

독립성을 유지하며 변경될 요소에 대한 편집이 가능하므로 효율적인 데이터 처리가 가능하게 된다. 또한 크기가 작은 업데이트 데이터만으로도 업데이트가 가능하므로 기존 방안보다 서버의 자원도 적게 사용하게 될 것이며, 텍스트로 구성된 장면 구성 정보를 Binary로 변환하여 전달되어야 하는 스트리밍 환경의 복잡한 장면에서도 자원 이용에 소비되는 시간을 크게 단축할 수 있다.

IV. 실험 결과

본 실험은 DOM update를 이용한 장면 구성 정보 변경 시스템을 구현하여 복잡도의 감소와 시간 단축에 대한 증명을 목적으로 하고 있다. 이는 640*480 해상도의 이미지를 기술하고 있는 XML 문서를 입력 받아 메모리상에서 DOM tree를 생성하고 Renderer에서 초기 이미지를 출력한다. 이후 DOM update를 이용하여 메모리상에 존재하는 DOM tree의 요소에 대한 추가/제거/변경과 같은 편집 명령을 전송한다.

이 때 변경된 요소에 대한 정보만 분석함으로써 다른 중복되는 요소의 정보를 분석하는 시간과 연산에 대한 부담을 줄이고자 한다.

정확한 실험을 위해 초기 장면과 추가 및 변경될 이미지, 텍스트 객체들은 동일한 콘텐츠를 사용하였으며 장면 구성에 사용된 XML 문서는 표 1을 통해 나타내었다. 각 변경

(가) 초기 장면

(나) 이미지 & 텍스트 객체 추가

(다) 이미지 & 텍스트 객체 변경 및 추가

그림 9. 장면 구성 변경 실험 콘텐츠

Fig. 9. Test content of changed Scene information

방법에 따른 소요된 시간들을 기록하여 그림 10과 표 2를 통해 시간적 효율성을 비교할 수 있었으며 DOM update를 이용한 장면을 변경될 시 화면의 깜빡임 없이 자연스러운 전환의 효과를 확인할 수 있었다.

그림 9는 장면 구성의 변경 과정을 나타내고 있다. 서버는 HTML 문서나 XML문서와 같은 형태의 장면 구성 정보를 클라이언트에게 전달하여 그림 9-(가)와 같은 초기 장면 정보를 제공한다.

클라이언트가 장면을 이용하는 도중 서버에서 부가정보

를 요청할 시 서버는 부가적인 이미지와 텍스트 정보가 담긴 XML문서를 제공한다. 클라이언트는 해당 부가정보를 포함한 XML을 분석하여 그림 9-(나)에서와 같이 초기 장면을 유지한 채 등장인물의 이미지와 텍스트 정보창을 표시한다.

만약 서버의 부가정보 제공으로 기존의 부가정보 형태가 변경과 새로운 부가 정보가 추가될 시 그림 9-(다)에서는 그림 9-(나)에서 추가된 정보창의 크기와 형태가 변경됨과 동시에 새로운 부가 이미지와 텍스트 정보를 통해 캐릭터

표 1. 변환된 장면 구성 정보 XML 예

Table 1. Example of XML about changed Scene information

<pre><?XML version="1.0" encoding="UTF-8"?> <header> -header 자세한 기술 생략- </header> <body> <XMLns:ev="http://www.w3.org/2001/XML-events" XMLns:saf="urn:mpeg:mpeg4:SAF:2005" XMLns:xlink="http://www.w3.org/1999/xlink" XMLns="http://www.w3.org/2000/svg" XMLns:smil="http://www.w3.org/2005/SMIL21" XMLns:lsr="urn:mpeg:mpeg4:LASer:2005"> ㉑ </body> </XML></pre>	
(가) 초기 장면 구성 정보 생성을 위한 XML	
<pre><?XML version="1.0" encoding="UTF-8"?> <header> -header 자세한 기술 생략- </header> <body> -body 중복 기술 생략- ㉑-1 </body> </XML></pre>	<pre> ㉑-2</pre>
(1) 기존 방안	(2) 제안 방안
(나) 이미지 & 텍스트 객체 추가를 위한 XML	
<pre><?XML version="1.0" encoding="UTF-8"?> <header> -header 자세한 기술 생략- </header> <body> -body 중복 기술 생략- ㉑-1 <imgid="SheldonInfo" src="SheldonInformation1.jpg"x="0" y="200" width="320" height="240"/> ㉑-1 </body> </XML></pre>	<pre> ㉑-2 <imgid="SheldonInfo"src="SheldonInformation1.jpg"x="0" y="200" width="320" height="240"/> ㉑-2</pre>
(1) 기존 방안	(2) 제안 방안
(다) 이미지 & 텍스트 객체 변경 및 추가를 위한 XML	

소개가 이루어진다.

이와 같이 장면 구성의 변경에 이용된 XML 문서는 다음 표 1과 같다.

표 1 (가)의 XML 문서는 크게 Header와 Body 부분으로 나누어져 있다. Header 부분에는 초기 객체 서술자의 프로파일 정보 및 복호화 정보를 기술하며 Body 부분에는 초기 장면 이미지 소스 정보(㉠)를 분석하여 초기 장면을 생성한다.

기존의 장면 구성 변경 방식으로 그림 9-(나)와 같은 초기장면에 이미지와 텍스트 요소를 추가할 경우, 표 (나)-(1)의 XML 문서를 이용하여 변경될 이미지와 텍스트 객체(㉡-1)의 정보와 표 (가)의 내용을 중복하여 생성과 분석 과정을 거치게 된다.

하지만 DOM method의 createElement()를 이용하여 장면을 변경시킬 경우 표 (나)-2와 같이 변경될 이미지와 텍스트 객체(㉡-2)의 추가하고자 하는 요소 정보만으로도 이미지와 텍스트를 추가시킬 수 있다.

표 1-(다)-(1)은 기존의 이미지와 텍스트 객체 정보의 변

환(㉢-1)과 새로운 캐릭터 소개 이미지와 텍스트 객체(㉣-1)를 추가하고자 하는 정보를 담고 있다. 이 역시 기존의 이미지와 텍스트 객체 정보의 변환(㉢-2)과 새로운 캐릭터 소개 이미지와 텍스트 객체(㉣-2)를 추가하고자 하는 정보만으로 중복된 정보를 제거하고 DOM method의 replaceChild()와 createElement()를 사용시 편집될 요소 정보만으로도 객체 추가 및 변경이 가능하다.

그림 10과 표 2는 기존의 방안과 제안된 방안을 사용함으로써 장면이 변경되는데 걸리는 시간을 비교한 실험 결과를 나타낸다. 1-1의 경우 두 실험 모두 초기 장면 구성

표 2. 장면 구성 변경 방안 비교 실험 결과 평균 데이터(Unit : Sec)
Table 2. Average time consumed between traditional and proposed methods

장면 구성 정보	기존 장면 구성 정보 변경 방안	DOM update를 이용한 장면 구성 정보 변경 방안
1-1	0.116	0.115
1-2	0.115	0.078
1-3	0.116	0.075

그림 10. 장면 구성 변경 방안 비교 실험 결과 데이터

Fig. 10. Result of Comparing Scene Composition update methods

정보를 분석한 뒤 DOM tree를 생성함으로써 분석에 걸리는 시간에 대한 큰 차이는 존재하지 않는다. 하지만 기존 변경 방식으로 매번 장면이 변경될시 새로운 XML문서를 파싱하여 DOM tree를 형성함으로써 전환되는데 걸리는 시간은 평균 0.116초의 결과가 도출되었다. 하지만 제안하는 방식으로 장면을 전환 될 시 소비된 시간은 0.077초로 약 52%의 시간적 단축률을 관측할 수 있었다. 이 실험에서 보여주는 평균 0.039초의 단축률은 초당 24~30장 장면을 사용하는 일반 디지털 방송에서 시간적으로 큰 단축률을 제공할 뿐만 아니라 깜빡임 없이 자연스러운 장면 전환이 가능한 효과를 나타낼 수 있다.

V. 결 론

리치미디어 서비스를 통해 시청자는 기호에 맞는 멀티미디어 콘텐츠를 선택하여 개인화된 데이터의 활용이 가능하다. 하지만 휴대폰과 같은 단말기와 제한된 네트워크 환경에서 기존의 리치미디어 서비스의 장면 변경을 지원할 시 시스템적인 측면에서 많은 연산량이 요구된다. 이와 같은 연산들은 단말 내에 의도하지 않은 버퍼링 문제를 발생 시킴으로써 원활하지 못한 서비스의 한계를 초래하게 되는 원인이 될 수 있다.

본 논문에서는 장면 구성 변경 방식을 웹페이지 변경 규격인 DOM update Method를 활용함으로써 보다 효율적인 리치미디어 서비스에 대한 가능성을 제시하며 화면전환에 있어 자연스러운 변경의 효과를 확인할 수 있었다.

이는 다양한 단말기에서도 장면 기술에 대한 연산에 필요한 시간을 대폭 줄임으로써 기존의 장면 구성 변경 과정에서 발생하는 서비스 지연과 오작동 문제를 사전에 예방한다. 또한 실험을 통해 기존 방안보다 장면 구성 변경에 필요한 정보의 양을 줄이고 분석에 소비되는 시간적 단축

효과를 검증하면서 원활한 리치미디어 서비스를 제공 가능성을 확인할 수 있었다.

이 방안을 적용시킴으로써 사용자와 소비자는 다양한 단말 환경에서 보다 편리하고 효율적인 리치미디어 서비스의 이용이 가능하며, 장면 기술 서비스의 진입 장벽을 낮춰 서비스를 더욱 활성화 할 수 있다는 장점이 있다. 이처럼 점차 발전되는 리치미디어 콘텐츠 기술은 이제 단순한 AV 중심 미디어 서비스에서 벗어나 보다 실용적인 양방향 리치미디어 콘텐츠 소비를 가능하게 하였다. 가요채널, 스포츠 같은 양방향 문화 콘텐츠와 원격 교육과 같은 진보적인 대화형 리치미디어 어플리케이션으로의 이용이 가능하게 되면서 기존의 서비스들과는 차별화되는 프리미엄 콘텐츠 서비스를 제공함으로써 새로운 부가가치 창출을 기대할 수 있다.

참 고 문 헌

- [1] ISO/IEC 14496-1:1999 Information Technology - Coding of audio-visual objects - Part 1 : Systems.
- [2] ISO/IEC 14496-20:2006 Information Technology - Part 20 : Light-weight Application Scene Representation(LASer) and Simple Aggregation Format(SAF)
- [3] W3C Extensible Markup Language (XML) 1.0 (Fifth Edition) - <http://www.w3.org/TR/2008/REC-XML-20081126/>
- [4] W3C Document Object Model(DOM) Level 3 Load and Save Specification version 1.0 - <http://www.w3.org/TR/2004/REC-DOM-Level-3-LS-20040407/>
- [5] KimByungchul et al. "An Efficient Transmission Method of Panoramic Multimedia Contents in a Limited Bandwidth Environment" JBE, Vol.16, NO.5, pp.811~823, Sep. 2011
- [6] LeeYunJu "A Scene Composition by Object Priority Order on MPEG-4 Player", kiise, 2003.4 vol.30(3)
- [7] LeeInjae et al. "Rich Media Framework based on MPEG-4 LASer PMSI Technique", JBE, Vol.15, NO.2, pp.248~264, March. 2010
- [8] W3C Scalable Vector Graphics(SVG) Tiny 1.2 Specification - <http://www.w3.org/TR/SVGTiny12/>
- [9] "DOM Scripting : Web Design with JavaScript and the Document Object Model", Jeremy Keith, 76p

저 자 소 개

김 규 현

- 1989년 2월 : 한양대학교 전자공학과 공학사
- 1992년 9월 : 영국 University of Newcastle upon Tyne 전기전자공학과 공학석사
- 1996년 7월 : 영국 University of Newcastle upon Tyne 전기전자공학과 공학박사
- 1996년~1997년 : 영국 University of Sheffield, Research Fellow
- 1997년~2006년 : 한국전자통신연구원 대화형미디어연구팀장
- 2006년~현재 : 경희대학교 전자정보대학 교수
- 주관심분야 : 영상처리, 멀티미디어통신, 디지털 대화형방송

박 정 욱

- 2010년 9월 : 경희대학교 전자전파공학 학사
- 2013년 2월 : 경희대학교 전자전파공학 석사
- 주관심분야 : 멀티미디어, MPEG-DASH, MPEG-LASer

김 병 철

- 2008년 2월 : 경희대학교 전자공학과 학사
- 2010년 2월 : 경희대학교 전자전파공학과 석사
- 2010년 3월 ~ 현재 : 경희대학교 전자전파공학과 박사과정
- 주관심분야 : 멀티미디어, MPEG 장면 구성 기술