

HTML5 기반 융합 서비스의 서비스이동 메커니즘 설계

최헌희[†], 송은지^{**}, 김근형^{***}, 김화숙^{****}, 조기성^{*****}

요 약

최근, W3C에서는 웹을 통해 다양한 웹 응용을 제공할 수 있는 기능을 포함하는 HTML5의 표준화가 추진 중이다. 광대역 무선 네트워크의 등장 및 스마트 TV, 태블릿 PC, 스마트 폰과 같은 스마트 단말이 웹 브라우저를 탑재함에 따라 스마트 단말을 통해 언제 어디서나 웹 서비스를 이용하는 것이 가능해졌다. 또한 사용자가 다수의 단말을 보유함에 따라 상황에 맞는 단말을 사용하여 웹 응용을 이용하는 멀티스크린 서비스의 요구가 증가하고 있다. 본 논문에서는 HTML5 기반의 웹 플랫폼에서 웹 객체를 그룹화하는 방안, 이동할 웹 객체의 정보를 추출하는 방안 및 다른 단말로부터 전달받은 웹 객체 정보를 기반으로 새로운 객체를 생성하는 방안을 제안한다. 또한 HTML5 기반 개방형 웹 플랫폼에서 단말 간 웹 응용을 이동하는 웹 서비스 이동 구조를 제안하였으며 이를 구현하여 테스트용 웹 문서와 일반 웹 문서에 대해 크롬 확장기능을 이용하여 검증하였다. 마지막으로 node.js와 웹 소켓을 이용하여 단말 간 객체 정보를 전달하는 메커니즘을 구현하고 제안한 방안을 검증하였다.

A Design of Service Migration Mechanism in HTML5-based Convergence Service

Hun-Hoi Choi[†], Eun-Ji Song^{**}, Geun-Hyung Kim^{***},
Hwa-Sook Kim^{****}, Ki-Seong Cho^{*****}

ABSTRACT

Recently, the W3C has developed the HTML5 standard which gives the basis for providing various web applications on the web environments. Because of the advent of the smart devices and the broadband wireless network, users can access the web applications on the smart devices at anytime and anywhere. In addition, the demand on the multiscreen services, which enables users to use the appropriate device to their situation, has increased, since users have various smart devices. In this paper, we propose the grouping mechanism of web objects on the HTML5 based web platform, the extraction mechanism of the web object information which is used to create the web object on other devices, and the web object creation mechanism based on the received web object information. In addition, we propose the web service migration architecture between devices on the open web platform and implement the grouping, extraction and creation mechanism of the web objects on the test web document and generic web document with Chrome extension. Finally, we implement the delivery mechanism of the web object information between devices using the node.js and the WebSocket technologies.

Key words: Web Service Extraction(웹 서비스 추출), Web Service Migration(웹 서비스 이동), Web Service Grouping (웹 서비스 그룹핑), HTML5

※ 교신저자(Corresponding Author): 김근형, 주소: 부산광역시 부산진구 가야3동 산학협력관 605호(614-714), 전화: 051)890-2271, FAX: 051)890-2265, E-mail: geunkim@deu.ac.kr

접수일: 2011년 12월 30일, 수정일: 2012년 2월 1일
완료일: 2012년 2월 15일

[†] 준회원, 동의대학교 디지털미디어공학과
(E-mail: hunhoi2203@gmail.com)

^{**} 준회원, 동의대학교 디지털미디어공학과
(E-mail: sej1272@naver.com)

^{***} 정회원, 동의대학교 영상정보공학과

^{****} 정회원, 한국전자통신연구원
(E-mail: hwskim@etri.re.kr)

^{*****} 정회원, 한국전자통신연구원
(E-mail: chokis@etri.re.kr)

※ 본 연구는 방송통신위원회의 차세대통신네트워크 원천 기술개발사업 (KCA-2011-09913-05002) 및 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초 연구사업(No.2011-0027080)의 지원으로 수행되었음.

1. 서 론

광대역 유무선 통신망의 등장과 스마트폰, 스마트 TV, 태블릿 PC와 같은 다양한 스마트 단말이 등장함에 따라 사용자가 언제, 어디서나 멀티미디어 통신 서비스를 사용하게 되었다. 또한 모든 스마트 단말에 웹 브라우저가 탑재되어 다양한 웹 서비스를 다양한 이용할 수 있게 되었다. W3C(World Wide Web Consortium)에서 표준화하고 있는 HTML5에서는 여러 처리 방법을 상세하게 규정하고 있어 모든 브라우저에서 일관된 UI(User Interface)를 제공할 수 있도록 하며, 어도비사의 플래시와 마이크로소프트사의 실버라이트를 사용하지 않고도 HTML(Hyper-Text Markup Language), CSS(Cascading Style Sheet), JavaScript를 이용하여 브라우저에서 동작하는 웹 어플리케이션을 만들 수 있는 기반을 제공하고 있다. 이를 위해서 HTML5에서는 멀티미디어 재생, 게임 및 이미지 편집에 필요한 API(Application Program Interface), 다양한 단말 기능을 사용할 수 있도록 하는 API, 모바일 환경을 고려한 오프라인 웹 어플리케이션 API, 웹 소켓, 웹 워커와 같은 웹 어플리케이션 개발에 필요한 API와 웹 서버와 독립적인 웹 어플리케이션 개발에 필요한 기능을 정의하고 있다. 최근에는 웹 브라우저의 피어간 직접 실시간 오디오/비디오 통신을 위한 클라이언트 API 표준도 준비 중이다.

인터넷에 접속되어 TV 상에서 인터넷 기반 서비스를 이용할 수 있는 커넥티드 TV(connected TV)와 IPTV(Internet Protocol TV) 서비스의 등장으로 단방향 방송 서비스의 시청뿐만 아니라 방송 콘텐츠에 대한 투표, 방송 콘텐츠와 SNS(Social Network Service)와의 연동과 같은 다양한 양방향 방송 서비스가 가능해졌다. 방송과 통신의 경계가 허물어지면서 콘텐츠 사업자, 통신 사업자, 포털 사업자 및 단말 제조사까지 방송 서비스에 다양한 데이터 부가서비스가 융합된 신개념의 방송 서비스를 단말에 제약 없이 동일하게 제공하려고 노력하고 있다. 즉 사용자의 다양한 스마트 단말의 스크린을 하나의 공유 스크린처럼 사용하며 하나의 콘텐츠를 다양한 단말에서 시청하는 멀티스크린 서비스에 대한 관심이 높아지고 있다.

멀티스크린 서비스의 가장 단순한 예는 각 단말의

스크린사이즈, CPU속도, 메모리, 코덱, 네트워크 인터페이스 및 속도 등 서로 다른 성능 요소를 가진 단말에 동일한 콘텐츠를 제공하는 것이다. 이러한 멀티스크린 서비스는 사용자가 자신이 처한 상황에 따라 사용 가능한 단말을 이용하여 콘텐츠를 이용할 수 있도록 한다. 이러한 멀티스크린 서비스 형태를 OSMU(One Source Multi Use)[1]로 분류하고 있다. 한 사용자가 여러 종류의 단말을 보유하게 되고 OSMU가 가능해지면서 집에서 TV를 이용하여 시청하던 콘텐츠를 외출 시 모바일 단말을 이용하여 연속해서 볼 수 있는 서비스로 확장이 필요하다. 이를 위해서 사용자의 프로파일을 이용한 서비스 이동 기술이 필요하다.

커넥티드 TV 보급과 IPTV 서비스가 고도화됨에 따라 방송 콘텐츠의 모습이 단일 콘텐츠 형태에서 방송과 관련 콘텐츠가 융합된 형태로 진화하고 있다. 이와 같은 융합서비스를 성능이 다른 다양한 단말에서 동일한 형태로 제공할 경우 문제가 발생한다. 예를 들어 데스크탑 PC를 대상으로 개발된 웹 콘텐츠를 화면 크기가 작은 스마트폰이나 태블릿 PC에서 표시하면 데스크탑 PC와 동일한 사용자 이용 경험을 제공할 수 없다. 또한 방송 통신 융합서비스와 관련하여 TV에서 방송과 웹 콘텐츠를 융합하는데 필요한 웹 표준의 검토가 W3C의 Web and TV IG (Interest Group)[2]에서 이루어지고 있다. 이처럼 방송과 다른 유형의 콘텐츠가 융합된 서비스를 하나의 단말에 모두 표시하기 위해서는 서비스를 이용하는 단말에 적합한 형태로 변환하거나 여러 단말의 특성에 맞도록 사용자가 이용단말을 선택하여 단말 간 서로 다른 유형의 서비스를 연동형으로 제공하는 것이 필요하다. 예를 들어 TV를 통해서 방송을 시청하면서 방송과 관련된 촬영 장소, 드라마 소품(자동차, 가구 등)관련 상품 정보 등을 스마트폰 또는 태블릿 PC에 표시한다. 스마트폰이나 태블릿 PC가 존재하지 않은 경우에는 부가정보가 TV 화면에 방송과 함께 표시될 수 있다. 이러한 멀티스크린의 형태를 ASMD(Adaptive Source Multi Device)[3]로 분류하고 있다.

다양한 형태의 스마트 단말로 구성된 미래의 멀티미디어 유비쿼터스 통신 환경에서는 현재의 멀티스크린 서비스뿐만 아니라 다양한 형태의 멀티스크린 서비스가 요구될 것이다. 다양한 스마트 단말 간의

스크린 공유 및 연동 형태의 멀티스크린 서비스를 지원하기 위해서는 개발한 서비스가 운영체제에 관계없이 스마트 단말에서 일관되게 실행되고 일관된 UI를 제공할 수 있는 환경이 요구된다. W3C에서는 언제 어디서나 다양한 단말 간 협업 및 연계를 할 수 있는 멀티스크린 서비스를 위해 웹 기술 중심으로 개방형 웹 플랫폼(Open Web Platform)[4]을 표준화하고 있다. 미래의 융합 서비스의 모습도 다양한 형태의 웹 기반 콘텐츠로 구성될 것이 예측되고 있다. 이에 따라 사용자가 융합 서비스를 효율적으로 이용하고 각자의 콘텐츠를 다른 사람과 공유하기 위해서는 융합 서비스를 구성하는 요소 서비스의 분리 및 결합 이동은 필수적이다. 본 논문에서는 멀티스크린 서비스를 구성하는데 기반이 될 웹 객체(웹 서비스¹⁾)의 분리 및 결합 이동 구조를 HTML5 기술에 기반하여 설계한 내용과 구현한 결과를 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 웹 환경에서 웹 객체를 선택하고 저장하며 공유하는 기존 연구에 대해서 살펴보고 3장에서는 본 논문에서 제안하는 웹 기반 융합 서비스의 각 웹 객체를 다른 단말로 이동하는 구조에 대해서 설명한다. 마지막으로 4장에서 결론을 맺는다.

2. 관련 기술

웹을 통해 다양한 정보 검색과 서비스를 이용하는 사용자는 필요에 따라 관련 URL(Uniform Resource Location)을 즐겨찾기에 저장해 놓거나 또는 이용한 URL을 기억하여 직접 주소입력을 통해 다시 이용한다. 이러한 형태는 사용자가 원하는 정보가 웹 문서의 일부분인 경우에도 웹 문서 전체를 다운로드하게 되어 네트워크 자원을 소모하게 된다. 또한 즐겨찾기 목록이 삭제되거나 정보 또는 서비스의 URL을 기억하지 못할 경우 사용자는 필요한 정보를 다시 검색하여야 한다. 이러한 문제를 해결하기 위해 웹 문서의 일부 객체를 저장하는 서비스가 제안되었다.

각 서비스는 객체 선택 및 저장 방법, 저장된 객체 제공 형태, 개인화된 웹 문서 제공 방법, 지원하는 브라우저와 플랫폼 측면에서 차이점을 가진다. 현재 모든 서비스는 객체를 서버 또는 사용자 단말에 저장

하는 형태로 제공하며 일부 서비스는 저장된 객체를 공유할 뿐 사용자 간 서비스 객체를 이동에 대한 연구는 많이 이루어지지 않았다. 최근 모바일 IPTV 서비스의 이동에 대한 연구[5]와 HTML5 기반 스트리밍 서비스의 단말 간 이동[6,7]에 대한 연구가 이루어졌다. 또한 실제 데이터를 다운로드 받는 방식이 아닌 목록 또는 URL만 가져와서 사용자가 요청하면 다시 로드하는 방식으로 서비스한다.

웹 객체 정보의 저장 방법은 크게 2가지로 분류된다. 첫 번째는 사용자의 계정을 기반으로 모든 정보를 웹서버에 저장하는 방법이다. 이 방법에는 Wepware[8]와 Memonic[9]이 있다. 두 번째는 사용자의 단말에 객체 목록을 저장하고 객체 요청이 발생하면 저장된 목록을 기반으로 웹 문서에서 객체를 로드하는 방법이다. 이 방법에는 Web slice[10,11], Fire clip[12], Arbitrary web slice[13], Web chunk[14]가 있다. 첫 번째 방법은 객체 정보를 서버에 저장하기 때문에 다른 사용자와 공유가 가능하며 두 번째 방법은 웹 서버로부터 객체 정보를 획득하는데 사용된다.

Wepware는 마이크로소프트의 인터넷 익스플로러, 구글의 크롬, 모질라의 파이어 폭스에서 브라우저의 확장 기능으로 제공된다. 이는 위젯 형태로 제공되며 관심 있는 웹 객체를 저장소에 저장한다. 저장한 객체는 개인화 페이지를 통해 객체의 레이아웃을 자유롭게 설정하여 볼 수 있다. 또한 인터넷 상의 Youtube 또는 Flickr의 비디오와 사진도 저장하는 메뉴를 제공한다. 저장할 객체의 선택은 웹 문서에서 마우스의 드래그(drag)로 이루어진다. 선택된 영역을 저장하여 보여줄 때는 웹 문서 전부를 로드하고 화면에는 사용자가 선택한 부분만을 보여준다. 관심 있는 영역은 일부분이지만 웹 문서 전부를 로딩하기 때문에 불필요한 네트워크 자원의 낭비가 발생한다. Memonic은 웹 문서 내의 객체를 그룹화하여 저장한다. 브라우저뿐 아니라 Mac, Window 시스템에 설치할 수 있도록 응용 프로그램도 제공하고 있으며, 아이폰, 안드로이드와 같은 스마트폰에도 어플리케이션으로 제공한다. 멀티 플랫폼 지원을 위해 사용자 계정 기반으로 단말 간 저장한 객체의 동기화도 제공한다.

Web slice와 Web chunk는 RSS(Really Simple Syndication)서비스와 웹 객체 선택 서비스가 결합된 구조로 'hslice'라는 새로운 클래스 속성 값을 정의

1) 본 논문에서는 웹 서비스와 웹 객체를 같은 의미로 혼용한다.

하여 콘텐츠 제작자가 RSS 서비스를 제공하고자 하는 웹 조각을 div 태그와 태그의 클래스 속성 값으로 'hslice'를 이용하여 생성한다. 사용자가 이 웹 조각을 즐겨찾기에 저장해 놓으면 웹 조각내의 요소(element) 단위로 갱신된 정보를 수신한다. Web slice는 인터넷 익스플로러의 브라우저 내에 내장된 기능이며 Web chunk는 파이어폭스의 확장 기능으로 인터넷 익스플로러와 같은 서비스를 제공한다.

Fire clip은 파이어 폭스에서 제공하는 것으로 인터넷 익스플로러의 웹 조각과 유사하다. 특정 영역에 마우스 오버가 되면 선택할 객체를 음영 처리하여 사용자가 필요로 한 부분을 쉽게 선택하며 페이지 내에서 여러 객체를 선택할 수 있다. 선택된 객체는 사용자의 PC에 저장되며 다른 사용자 또는 다른 단말과의 공유는 불가능하다. 저장된 객체를 표시할 때 객체의 레이아웃을 설정할 수 있으며 객체를 포함하는 원본 페이지를 로드할 수 있는 기능도 제공한다. Arbitrary web slice는 크롬에서 제공하는 것으로 인터넷 익스플로러의 웹 조각과 유사하다. 이는 콘텐츠 제작자가 div 태그와 id를 이용하여 정의한 웹 문서의 일부분을 표현하며 크롬의 확장 기능으로 제공한다. 확장 기능에서 div와 id를 검색하여 콘텐츠 제작자가 지정한 내용을 팝업(popup) 페이지에 출력한다.

기존 연구는 웹 문서에서 개별적인 웹 객체를 선택하는 방법과 선택한 객체를 개인화하기 위해서 사용자의 PC 또는 네트워크 서버에 저장하는 것으로 웹 기반 융합 서비스를 분리 및 결합 이동하는 기술에 대한 연구는 미흡한 상황이다. 웹 기반 융합 서비스를 구성하는 웹 객체 중 관련성이 있는 것을 그룹화하여 이동할 객체의 선택 방안, 사용자의 판단에 의해 이동할 객체를 선택하여 그룹화하는 방안, 그룹화한 객체를 단말 간 이동할 수 있는 방안 및 단말 간 웹 객체 이동 구조에 대한 연구가 사용자의 이용 경험을 높일 수 있다는 점에서 필요하다.

3. 웹기반 객체 이동 구조

3.1 이동할 웹 객체 식별 방법

웹 문서는 태그(tag) 이름으로 구분되는 다양한 요소(element)로 구성된다. 브라우저는 태그 이름에 따라 태그 내의 정보(콘텐츠)를 적절하게 표현한다. 그러므로 태그 이름은 웹 문서를 이루는 웹 객체의 속

성을 구분하는데 사용한다. 그러나 하나의 웹 문서에 같은 태그 이름을 가지는 웹 객체가 다수 존재할 수 있기 때문에 태그만을 이용하여 이동할 웹 객체를 식별하는 것은 불가능하다. 각 태그는 class, id, style, title을 비롯한 여러 속성(attribute)을 가지고 있다. id 속성 값은 웹 문서의 요소를 식별하기 위해 사용되어 웹 문서 내에서 유일한 값을 가진다. class 속성 값(이하 클래스 이름이라 사용)은 웹 문서의 요소들을 그룹화하는데 사용된다. id 속성과 클래스 속성 값은 스타일 시트 내의 class와 id와 연관을 맺는데 사용된다.

본 논문에서는 이동할 웹 객체 하나를 식별하기 위해서는 태그 이름과 id 속성 값을 사용하고 관련된 여러 웹 객체를 하나의 그룹으로 식별하기 위해서는 태그 이름과 id 속성 값에 클래스 이름을 결합하여 사용할 것을 제안한다. 클래스 이름으로 접두어 'sem_'이 포함된 클래스 이름을 사용하여 동시에 이동될 수 있는 그룹을 구성한다. 'sem_' 접두어로 시작하는 클래스 속성 값을 사용하여 웹 객체를 그룹핑하여 함께 이동할 웹 객체를 식별한다. PC에서 웹 객체를 식별하기 위해서 마우스 클릭 시 생성되는 이벤트 객체(event)의 srcElement 객체를 통해서 클래스 속성 값, 태그 이름, id 속성 값을 얻는다.

이와 같은 방법을 통해 웹 문서 내에서 이벤트 객체로 식별 가능한 객체를 표 1에 나타냈다. 웹 문서 내에 여러 객체가 존재하는데 그 중 사용자가 이동을 요구할 가능성이 있는 객체를 중심으로 정리했다.

표 1에 표현한 객체는 이벤트 객체로 식별이 가능한 객체로 실제 단말 간 객체 이동의 기본 단위는 아니다. 객체가 이동될 때 객체 내부에 속한 정보를 추출하기 위해 필요한 객체로서 의미만을 가진다.

3.2 단말 간 웹 객체 이동을 위한 정보

이동시킬 웹 객체를 식별 선택한 이후 다른 단말에서 이용하기 위해서는 다른 단말에서 웹 객체를 생성하는데 필요한 정보가 전달되어야 한다. 전달되어야 할 정보는 객체의 특성에 따라 다르다. 웹 객체가 가지는 정보는 URL 정보 또는 텍스트 정보이다. 웹 객체가 URL 정보를 가지면 텍스트 정보를 갖지 않고 텍스트 정보를 가지는 경우는 URL 정보를 갖지 않는다. 텍스트 정보를 갖는 객체는 객체의 value 속성 값과 innerHTML 속성 값으로 표현되는 것으

표 1. 이벤트 객체로 식별 가능한 객체

태그 이름 (객체 이름)	객체 특징
button	푸시 버튼으로 텍스트나 이미지 삽입 가능
form-input	사용자 정보를 여러 타입에 따라 선택
image	웹에 있는 이미지 파일 디스플레이
select-option	여러 개의 옵션으로 리스트 구성
textarea	웹 문서 내에 여러 줄의 텍스트 입력 가능
video	웹 상의 비디오 파일
audio	웹 상의 오디오 파일
div	HTML문서를 분할, 스타일 적용을 위한 블록화
p	HTML 문서의 문단을 나눔
iframe	하나의 웹 브라우저에 여러 HTML문서 표시
ol, il	순서 있는/없는 리스트 구성
li	리스트 항목 정의
hgroup, h1~h6	여러 개의 제목(h1~h6)을 하나로 정의

로 분류된다. 그러므로 웹 객체가 가지는 주요 정보는 src 속성 값, value 속성 값, 또는 innerHTML 속성 값 중 하나이다.

표 1의 객체 중 image, audio, video, iframe 객체는 해당 콘텐츠의 URL 정보가 각 태그의 src 속성 값에

포함되어 있다. 그러므로 이 객체들의 주요 정보는 URL 정보이다. textarea와 form-input 객체의 정보는 value 속성 값으로 표현된다. innerHTML 속성 값으로 표현되는 객체는 div, p, ol, ul, li 이다. innerHTML 속성 값을 사용하는 객체들은 대부분 시작태그(< >)와 끝태그(</ >)가 존재한다. 따라서 텍스트로 이루어져 있거나 다른 객체를 포함하는 객체를 함께 이동할 경우 포함되어 있는 내용을 모두 이동할 수 있기 때문에 innerHTML을 이용할 수 있다. 웹 객체를 다른 단말로 이동할 때 전송되어야 할 객체의 주요 정보와 부가 정보는 표 2와 같다.

3.3 이동할 객체 추출 및 생성

웹 문서를 구성하는 웹 객체 일부를 다른 단말로 이동하기 위해서는 이동할 웹 객체의 선택이 요구된다. 웹 문서를 구성하는 객체들은 모두 독립적인 경우도 있고 서로 관련이 있을 수도 있다. 관련성이 있는 웹 객체들이 있는 경우 다른 단말로 이동할 때 함께 이동시킬 필요가 있다. 이러한 관련성은 콘텐츠 제작자와 사용자가 명시적으로 결정할 수 있어야 한다. 또한 이동할 객체 선택 시 관련이 있는 객체를 한 번에 선택할 필요가 있다. 본 논문에서는 콘텐츠 제작자가 웹 문서를 생성할 때 관련이 있는 객체들을 하나의 그룹으로 구성하는 방법과 사용자의 선택에

표 2. 객체 별 이동 시 필요한 부가 정보

주요 정보	객 체 명	부 가 정 보
URL	image	
	iframe	
	audio	currentTime(현재 재생시점을 다른 단말기에서 보고 싶은 경우), audio를 제어하는 javascript코드
	video	currentTime(현재 재생시점을 다른 단말기에서 보고 싶은 경우), video를 제어하는 javascript코드
value	textarea	
	form-input(type button)	button을 눌렀을 때 실행되는 javascript 코드
	form-input(type reset)	
	form-input(type submit)	submit되는 문서의 URL
innerHTML	form-input(type text)	
	button	type(button, submit, reset), button을 눌렀을 때 실행되는 javascript 코드
	select-option	
	div, p, ol, ul, li	
	hgroup, h1~h6	

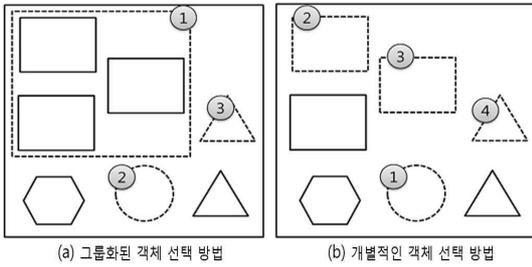


그림 1. 객체 그룹화 방법

따라 이동할 객체의 그룹을 구성하는 방법을 제안한다. 사용자는 제공자가 그룹화한 객체를 한 번에 선택하여 이동시킬 수 있다. 콘텐츠 제작자가 객체를 그룹화하여도 사용자가 관련이 있다고 생각되는 객체를 직접 선택하여 그룹을 생성한 후 이동시킨다. 본 논문에서는 웹 문서의 객체를 그룹화하기 위해 클래스 속성 값(이하 className)을 이용한다. 한 객체에는 여러 속성이 존재하는데 객체 정보를 가져올 수 있는 속성은 tag, id, class 이다. tag는 그룹을 만들 수 있는 속성 값을 지정해 줄 수 없다. id는 웹

문서에서 한 객체에 하나만 가질 수 있는 고유의 값이므로 그룹을 지정하기 위해 여러 객체에 같은 id 속성 값을 지정할 수 없다. class는 스타일 시트를 지정할 때 같은 스타일을 적용할 객체를 그룹화하는데 사용하는 것이나 한 객체가 클래스 속성 값으로 여러 className을 지정할 수 있기 때문에 웹 문서에서 함께 이동할 객체를 그룹화하는데 class 속성을 이용한다.

본 논문에서 제안하는 이동 객체 그룹화 방법을 그림 1에 나타냈다. 그림 1의 (a)는 콘텐츠 제작자에 의해 className으로 묶여있는 객체들을 선택한 뒤 추가적으로 객체를 선택하여 그룹화하는 방법이며, 그림 1의 (b)는 그룹화가 이루어져 있지 않은 웹 문서에서 사용자가 그룹화할 객체를 각각 선택하여 그룹화하는 방법이다. 두 방법 모두 선택한 객체를 모두 개별적으로 다시 해제할 수 있다.

웹 문서에서 이동할 객체를 추출하는 절차는 그림 2와 같다. 객체 상에서 마우스 클릭 이벤트가 발생하면 이동할 객체에 대해 그룹 이름이 설정되어 있는지

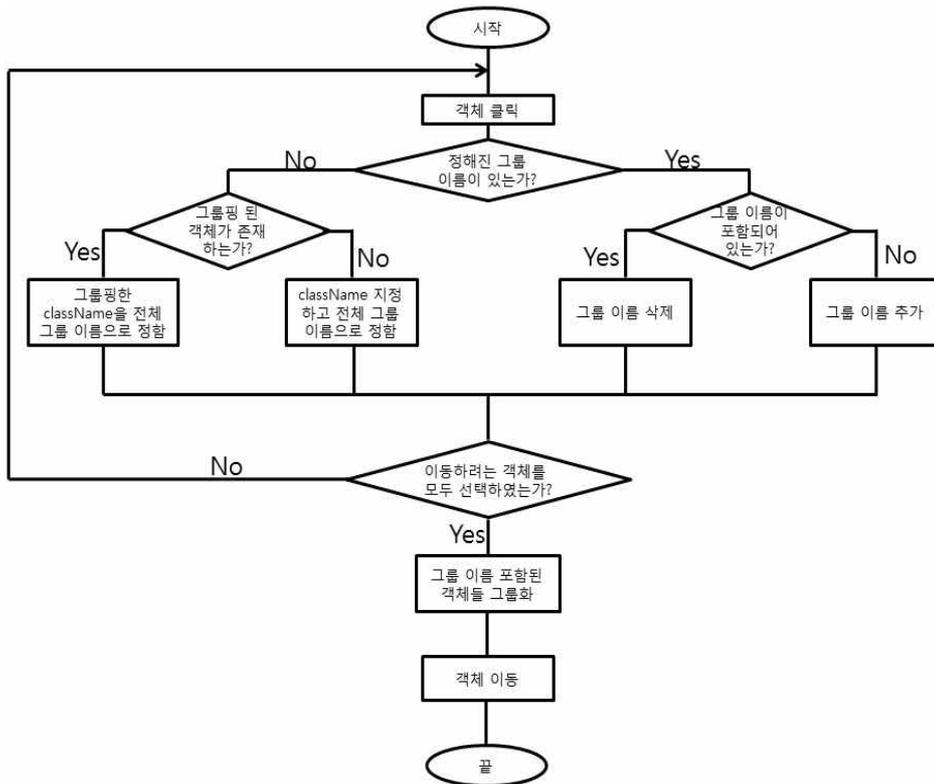


그림 2. 객체 추출 및 이동 절차

확인한다. 그룹 이름이 설정되어 있지 않으면(No) 왼쪽 절차를 수행하고 그룹 이름이 설정되어 있으면(Yes) 오른쪽 절차를 수행한다. 그룹 이름이 설정되어 있지 않은 상태에서 마우스 클릭으로 선택된 객체가 콘텐츠 제작자에 의해 그룹화된 것이라면 선택된 객체의 className중 'sem_' 접두어로 시작하는 className을 이동할 객체들의 그룹 이름으로 설정한다. 또한 그룹 이름이 설정되지 않은 상태에서 선택된 객체가 그룹화 되지 않은 객체이면 다른 객체와 중복되지 않는 임의의 className을 지정하여 그것을 그룹 이름으로 설정한다. 이후 객체가 선택된 후 함께 이동할 객체를 추가하거나 객체의 선택을 해지하는 과정이 오른쪽 절차이다. 그룹 이름이 설정된 후 객체의 선택이 이루어지고 선택된 객체의 className에 그룹 이름이 포함되지 않았으면 그룹에 추가할 객체로 판단하고 className에 그룹 이름이 포함되어 있으면 className에서 그룹 이름을 삭제하여 그 객체의 선택을 해지한다. 이처럼 함께 이동할 객체를 모두 선택한 이후에 설정한 그룹 이름으로 객체를 그룹화하고 이를 이동하기 위해 전달할 객체 정보를 추출한 후 JSON(JavaScript Object Notation) [15] 형식으로 패키징한다.

그룹 이름으로 그룹화된 객체를 이동하기 위해서는 그룹에 속한 웹 객체들의 객체 정보를 정확하게 추출하고 추출된 각각의 정보를 전송을 위해 메시지로 패키징하는 과정이 필요하다. 그림 3은 객체의 주요 정보인 URL, value, innerHTML을 기준으로 각

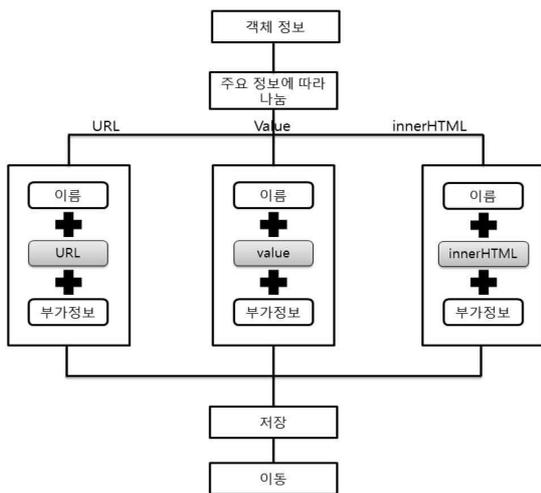


그림 3. 객체 주요 정보에 따라 객체 정보 구성 방법

각의 객체를 구분하고, 각 객체에 해당되는 태그 이름과 주요 정보와 부가 정보를 JSON 형태로 저장한 후 객체 정보 전달 모듈로 전달한다.

그림 4는 JSON 형태로 전달받은 메시지로 부터 객체의 이름을 추출해 낸 후 객체 이름을 기반으로 주요 정보와 부가정보를 차례대로 추출한다. 추출한 객체 이름을 기반으로 document.createElement() 메소드를 호출하여 태그에 해당하는 요소를 생성하고 주요 정보와 부가 정보를 이용하여 관련 속성 값을 설정한 후 document.body.appendChild() 메소드를 호출하여 새로운 객체를 웹 문서의 <body> 영역에 추가하여 웹 객체를 생성한다.

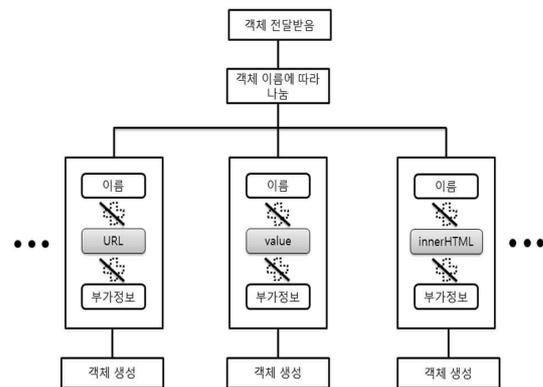


그림 4. 전달받은 객체 정보를 이용한 객체 정보 추출 절차

3.4 단말 간 객체 이동 구조

단말 간 객체를 이동하기 위한 기본 구조는 그림 5와 같다. 단말의 웹 브라우저 간 직접적인 통신의 어려움을 서버를 통해 해결한다. 또한 전체 사용자에 대한 정보 관리와 객체를 전달받는 사용자가 부재

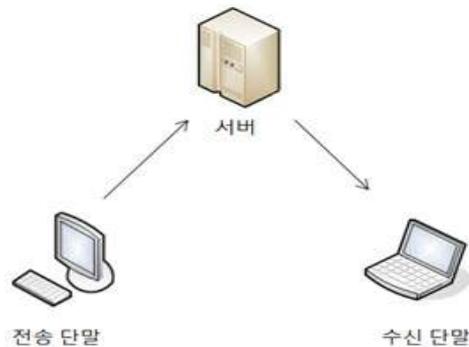


그림 5. 객체 전송을 위한 기본 구조

중 발생할 수 있는 전송 오류, 임시 보관 문제 등을 해결하기 위해 단말 간에 서버를 사용하는 것이 필요하다. 본 논문에서는 단말과 서버와 통신을 위해서 웹 소켓[16]을 사용하였으며 서버는 node.js[17]와 socket.io.js[18]를 기반으로 설계하고 개발하였다.

단말 간 객체 이동 방식은 전달하고자 하는 주체에 따라 푸시(push), 풀(pull), 제3자(3rd Party) 모드로 구분한다. 푸시 모드는 객체를 이용하는 단말에서 다른 단말로 객체의 이동을 요청하는 방식으로 가장 널리 사용된다. 풀 모드는 객체의 이동 요청이 이용하고자 하는 단말에서 현재 객체를 이용 중인 단말로 전달이 되어 객체가 전달되는 방식이다. 제3자 모드는 객체를 이용하는 단말과 객체가 이동되어야 할 단말이 제 3의 단말에서 객체의 이동을 요청하는 방식이다. 이 경우 제 3의 단말은 다른 단말의 객체 이용 상태에 대한 정보를 가지고 있어야 한다.

크롬 브라우저를 이용한 객체 이동 구조는 그림 6의 크롬 브라우저 확장 기능[19]의 기본 구조를 바탕으로 한다. 크롬의 확장기능은 사용자에게 보여지는 popup.html 문서, 확장 기능 뒤에서 실행되는 background.html 문서 및 문서와 연계하여 실행되는 contents_script.js파일로 구성된다. 크롬 확장기능 문서와 상호작용(interaction)에 대한 권한을 contents_script.js에게만 부여하고 있어 앞에서 설명한 웹 객체 식별 및 추출 기능을 contents_script.js에서 구현하게 된다. contents_script.js에서 추출된 객체는 background.html 또는 popup.html로 전달되며 background.html 문서는 contents_script.js에서 추출한 객체의 정보를 목표 단말에 전달하기 위해 전송 메시지로 패키징하고 서버로부터 전달되어온 메시지를 해석하여 새로운 객체를 생성하도록 contents_script.js로 메시지를 전달하는 역할을 한다. 또한 서

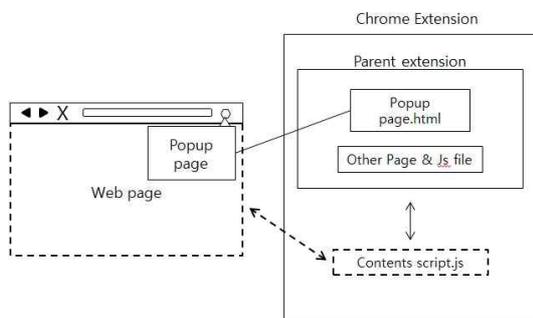


그림 6. 크롬 브라우저 확장기능의 기본 구조

버와의 통신 역할을 한다.

크롬 확장기능을 사용해 객체 추출 및 이동을 위한 구조는 앞서 설명한 기본 구조를 따르며, 그림 7과 같다. 객체 추출 및 생성 기능과 전달 및 수신되는 메시지에 대한 처리를 수행하는 객체 관리자(Object Manager)와 서버와 연결되어 객체 전달 및 수신 기능을 하는 에이전트(Agent)로 나뉜다. 객체관리자는 현재 사용 중인 웹 문서에 대해 처리하는 자바스크립트 코드로 항상 실행된다. 사용자가 이동 객체를 추출하고 선택된 객체에 대해 그룹화하는 작업을 수행한다. 그룹화한 객체는 서버로 전달을 위해 객체 정보를 에이전트에게 전달된다. 객체 관리자는 객체 추출 뿐 아니라 에이전트가 다른 사용자로부터 전달받은 객체 정보를 전달받아 해석하고, 웹 문서 상황에 따라 새로운 창 또는 현재의 웹 문서에서 전달받은 객체를 사용자가 볼 수 있도록 한다.

에이전트는 객체 관리자로부터 전달받은 객체 정보를 패키지(package)하여 서버로 전송하고 다른 사용자로부터 전달받은 객체 정보를 역 패키징(unpackage)하여 객체 관리자로 전달한다. 에이전트의 객체 정보의 패키징 및 역 패키징은 JSON 형식을 따라 서버로부터 받은 수신 객체에 대한 JSON형식의 정보를 역 패키징을 하여 객체 관리자로 전달하는 역할을 한다.

객체 관리자와 에이전트는 확장기능 내에서 상호간 정보 교환을 위해 크롬 확장기능의 메시지 전달 API를 사용한다. 메시지 전달 API는 메시지를 전달하는 chrome.extension.connect()의 postMessage() 메소드와 전달된 메시지를 수신하는 chrome.

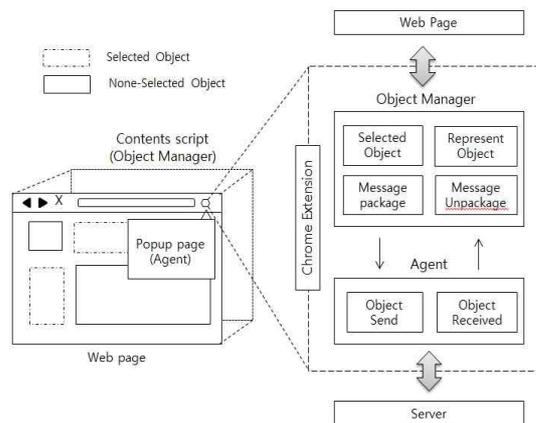


그림 7. 객체 추출 및 이동을 위한 Chrome 확장기능 구조

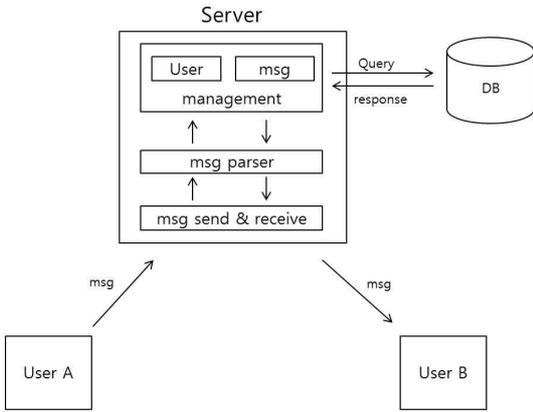


그림 8. 객체 이동을 위한 서버 기능 구조

extension.onConnect.addListener()로 나뉜다. 객체 관리자는 추출된 객체를 하나의 메시지로 구성하고 에이전트가 실행되면 메시지를 전달한다.

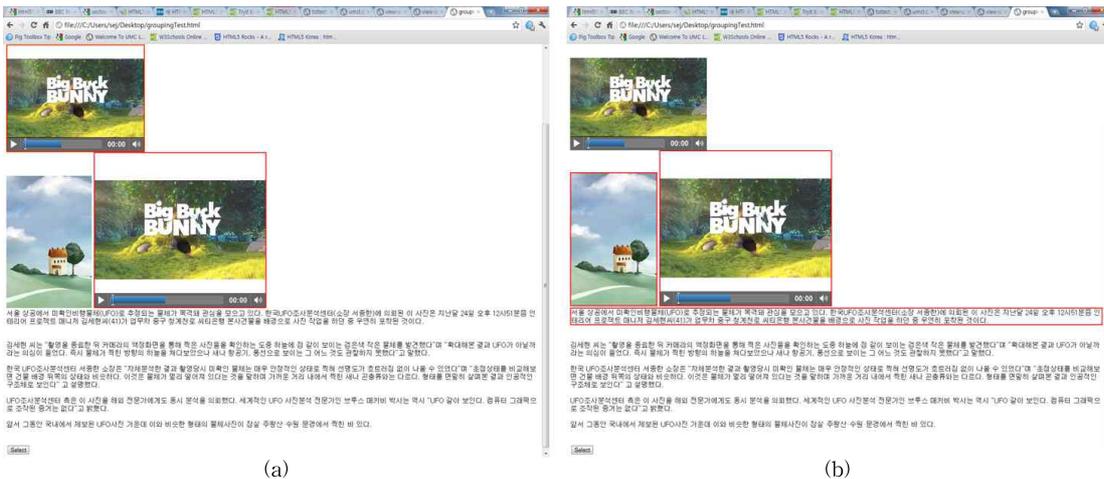
사용자 단말로부터 전달된 메시지를 처리하는 서버의 기능은 그림 8과 같다. 서버가 사용자로부터 전달받는 메시지의 종류는 5가지로 분류한다. 서버에게 사용자 목록을 요청하는 메시지, 객체 이동 메시지(push), 자신에게 전달된 메시지를 확인하는 메시지, 다른 사용자에게 객체 이동을 요청하는 메시지(pull), 사용자간 객체 이동 요청메시지(3rd party)가 있다.

사용자는 객체를 전달할 사용자를 선택하기 위해 서버에 현재 등록된 사용자 목록을 요청하는 메시지를 전송한다. 서버로부터 로그인 사용자 목록을 전달

받아 객체를 전달할 사용자를 선택한 뒤 추출된 객체와 함께 객체 이동 메시지를 통해 객체를 이동한다. 사용자가 서버에 재접속 할 경우 사용자는 서버에 자신에게 전달된 메시지가 있는지 확인하는 메시지를 전송한다. 서버는 전달된 메시지에 따라 사용자 아이디와 비밀번호를 데이터베이스에서 확인하고 전달된 메시지가 있을 경우 사용자에게 객체를 전달한다. 사용자는 다른 사용자 단말의 객체를 가져오기 위해 서버에게 객체 이동 요청 메시지를 보낼 수 있다. 제 3자가 서로 다른 사용자간의 객체 이동을 위해 서버에 객체 이동 요청을 하는 메시지가 있다. 서버는 사용자로부터 전달 받는 각 메시지를 msg parser를 통해 해석하고 해석된 메시지에 따라 각 기능을 실행한다.

앞에서 설계한 단말 간 웹 서비스 이동을 위해 개별 객체 또는 그룹화된 객체를 선택하여 객체 정보를 추출한 결과 화면은 다음 그림 9와 같다.

그림 9의 (a)는 웹 문서의 웹 객체 하나를 클릭하면 같은 className을 가지는 객체가 모두 선택된 것을 보인다. 이들 객체는 콘텐츠 제작자가 웹 문서를 만들 때 비슷한 의미를 가지는 객체들을 그룹화하여 같은 className을 지정한 경우이다. 이 때 사용자는 콘텐츠 제작자가 그룹화한 객체들을 한 번에 선택하여 이동시킬 수 있다. 콘텐츠 제작자가 웹 문서 제작 시 객체를 사전에 그룹화하여도, 사용자가 의미 있다고 생각되는 객체가 콘텐츠 제공자와 다를 수 있고, 그룹화된 객체가 아닌 사용자가 직접 선택하고



(a)

(b)

그림 9. 객체 식별 결과

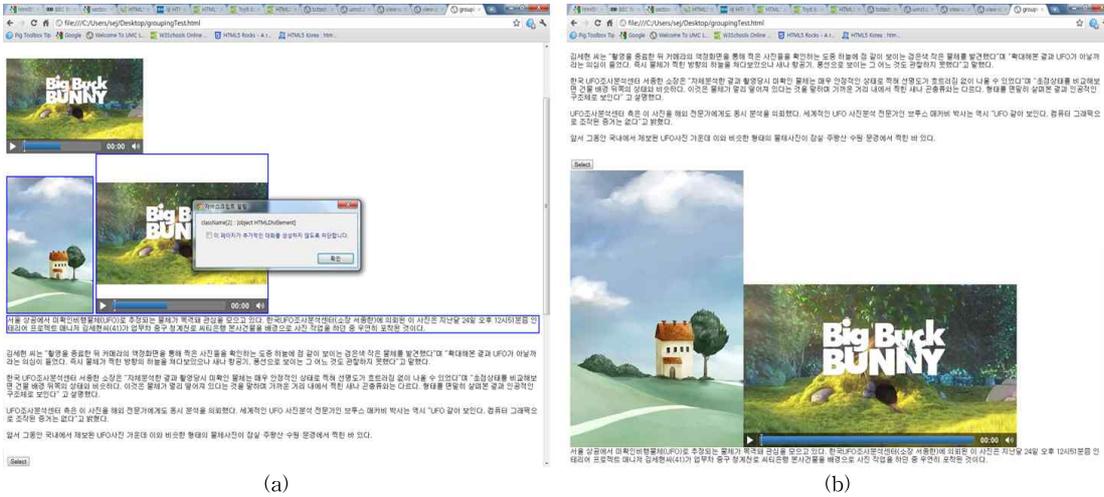


그림 10. 객체 정보 추출 및 정보에 대한 객체 생성 모습

그룹화하여 이동할 수 있다. 따라서 그림 9의 (b)와 같이 선택했던 객체중 하나를 취소하고 사용자가 이동하고 싶은 다른 객체들을 선택하여 그룹화할 수 있다.

그림 10은 선택한 객체들을 그룹화하여 객체 정보를 추출하고 정보에 따라 객체를 생성한 모습이다. 그림 10의 (a)는 이동하고 싶은 객체를 모두 선택하여 그룹화된 것으로, 그룹 내의 객체는 동일한 class-Name이 포함되어 있으므로 document.getElements-ByClassName() 메소드를 사용하면 웹 문서에 생성된 순서대로 객체 정보를 추출하여 배열에 저장한다. 그림 10의 (b)는 추출된 객체 정보로 실제 객체를 생성한 모습이다. 객체의 주요정보를 이용하여 객체가 생성되므로 전달되기 전 모습과는 다를 수 있다. 전달되기 전의 모습과 같은 모습으로 생성되기 위해서는 크기 정보를 부가정보로 저장하여 그룹화 할 때 함께 묶어 이동한다.

4. 결 론

본 논문에서는 HTML5 기반 융합 서비스의 서비스를 단말 간 이동시키기 위한 구조를 제안하였다. 특히 웹 문서에서 이동할 객체를 그룹으로 선택하는 방법과 선택한 객체들의 정보를 추출하는 방법, 단말 간 객체 정보의 전달 방법 및 전달받은 객체 정보를 기반으로 새로운 웹 객체를 생성하는 방법들을 제안하였다. 또한 테스트 웹 문서와 크롬 확장기능을 이

용하여 일반 웹 문서에 대해서 객체를 선택하고 추출하는 방법을 구현하였으며 웹 소켓과 node.js를 이용하여 서버기반 단말 간 웹 객체를 이동하기 위해 필요한 메시지 및 객체 정보의 전달 메커니즘도 구현하였다. 본 논문에서 제안한 방법은 다양한 웹 객체를 가진 웹 문서에서 자신에게 의미가 있는 객체만을 추출해 다른 단말에서 볼 수 있도록 함으로써 전체 웹 문서의 URL을 이용하여 웹 문서 전체를 다시 다운로드하는 방법보다 효율적임을 알 수 있다. 이는 웹 콘텐츠의 공유 작업 공간과 같이 웹의 정보 중 필요한 객체를 공유화면으로 이동시켜 공동 작업하는 서비스에도 적용 가능하다.

앞으로 HTML5 기반의 융합 서비스 이동을 통합적으로 관리할 수 있는 프레임워크에 대한 연구와 웹 앱에 대한 서비스 이동성에 대한 연구를 진행할 계획이다.

참 고 문 헌

[1] 김화숙, 이현진, 조기성, “N-Screen 서비스 현황 및 연구 개발 이슈,” 정보과학회, 제29권, 제7호, pp.9-15, 2011.
 [2] W3C Web and TV Interest Group website, <http://www.w3.org/2011/webtv/>, 2011.
 [3] 윤장우, 전승현, 이호연, 허경우, 이현우, 류원, “N-스크린 서비스 분류와 표준화 기술,” 정보과학회, 제29권, 제7호, pp.23-30, 2011.

- [4] W3C Open Web Platform wiki site, http://www.w3.org/wiki/Open_Web_Platform/, 2012.
- [5] 장지원, 김근형, “사용자와 서비스 세션 이동성 보장을 위한 모바일 IPTV 서비스 플랫폼 개발,” 한국디지털콘텐츠학회 논문지, 제10권, 제1호, pp. 87-96, 2009.
- [6] 김근형, “웹 서비스의 사용 환경 동기화 기술 및 웹 서비스 세션 이동 기술 분석,” 산업기술연구개발연구소, 산업기술연구지, 제24권, 제1호, pp. 63- 74, 2010.
- [7] 최현희, 김근형, “HTML5 기반 HTTP 스트리밍 환경에서의 서비스 이동성 연구,” 한국멀티미디어학회논문지, 제14권, 제7호, pp. 905-016, 2011.
- [8] Wepware website, <http://www.wepware.com>, 2012.
- [9] Memonic website, <http://www.memonic.com>, 2012.
- [10] Microsoft Web slice website, [http://msdn.microsoft.com/en-us/library/windows/desktop/cc304073\(c=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/cc304073(c=vs.85).aspx), 2012.
- [11] Microsoft Web slice website, [http://msdn.microsoft.com/en-us/library/cc196992\(VS.8-5\).aspx](http://msdn.microsoft.com/en-us/library/cc196992(VS.8-5).aspx), 2012.
- [12] Fire Clip website, <http://fireclip.awardspace.info>, 2010.
- [13] Arbitrary Web slices website, <http://code.cheesydesign.com/?p=450>, 2010.
- [14] Web chunk website, <https://addons.mozilla.org/en-US/firefox/addon/webchunks/>, 2008.
- [15] JSON website, www.json.org, 2012.
- [16] WebSocket website, <http://dev.w3.org/html5/websockets/>, 2012.
- [17] node.js website, <http://nodejs.org>, 2012.
- [18] socket.io.js website, <http://socket.io>, 2012.
- [19] Google Chrome Extensions website, <http://code.google.com/chrome/extensions/index.html>, 2011.



최 현 회

2011년 2월 동의대학교 영상정보 공학과 학사
2011년 3월~현재 동의대학교 디지털미디어공학과 석사과정
관심분야 : 멀티미디어 스트리밍, 웹 기반 융합 서비스 이동, 무선 네트워크



김 화 속

1991년 경북대학교 컴퓨터공학과 학사
2002년 충남대학교 컴퓨터공학과 석사
1991년 1월~현재 한국전자통신연구원 스마트서비스 제어 연구팀 책임연구원
관심분야 : 서비스제어기술, 서비스 이동성, IP기반 통합 제어



송 은 지

2012년 2월 동의대학교 영상정보 공학과 학사
2012년 3월~현재 동의대학교 디지털미디어공학과 석사과정
관심분야 : 웹, 웹 기반 융합 서비스 이동, 멀티미디어 스트리밍



조 기 성

1982년 경북대학교 전자공학과 학사
1984년 경북대학교 전자공학과 석사
현재 한국전자통신연구원 스마트 서비스제어연구팀 팀장
관심분야 : 서비스 제어 기술, 방송통신융합서비스, IPTV, All-IP통합망



김 근 형

1986년 2월 서강대학교 전자공학과 학사
1988년 2월 서강대학교 전자공학과 석사
2005년 2월 포항공과대학교 컴퓨터공학과 박사

1988년~1990년 LS 산전 연구소 연구원
1990년~1993년 삼성종합기술원 선임연구원
1993년~2007년 8월 (주) KT BcN 본부 수석연구원
2007년 9월~현재 동의대학교 영상정보공학과 조교수
관심분야 : 웹 기반 융합 서비스 이동, P4P 스트리밍, Content Centric Network