

이종 브라우저 간 HTML5 기반 서비스 이동기술 설계

송은지[†], 김근형^{**}

요 약

인터넷 사용이 우리 생활의 한 부분이 되었고 인터넷을 사용할 수 있는 단말이 증가함에 따라, 다양한 단말을 이용해 언제, 어디서나 웹 서비스를 제공받을 수 있게 되었다. 웹에 존재하는 콘텐츠는 여러 형태로 사용자에게 제공될 수 있고, 사용자는 콘텐츠를 사용자의 단말로 연속해서 이용하고자 한다. 웹 브라우저 확장기능은 기본적으로 브라우저에서 지원되지 않는 기능을 부가적으로 추가하는 기능으로 모든 브라우저들이 이 기능을 제공하고 있다. 확장기능을 통해 여러 브라우저에서 동일한 서비스를 제공할 수 있다. 본 논문에서는 우리가 연구한 웹 페이지의 객체를 식별하는 방법과 이동하기 위해 필요한 정보, 그리고 식별한 객체를 추출하고 이동한 뒤 생성되는 방법을 적용하여 이종 브라우저 간 객체 이동 구조를 제안한다. 이를 위해 대표적인 브라우저의 확장기능 구조를 분석하고 객체 이동 기능을 개발하기 위해 필요한 파일들에 대해 살펴본다. 또한 각 브라우저의 확장기능에 필요한 파일들이 서로 메시지를 주고받는 모습과 웹 페이지와 상호작용하는 방법에 대해 알아본다. 마지막으로 확장기능을 이용하여 웹 페이지에서 선택한 객체를 다른 브라우저로 이동하는 모습을 구현하였다.

A Design of HTML5-based Service Migration Technology between Heterogeneous Browsers

Eun-Ji Song[†], Geun-Hyung Kim^{**}

ABSTRACT

The Internet has become a part of our lives. As the number of devices with Internet accessibility increases, users can use web services with those devices anytime, anywhere. Web contents on the web page can be delivered to user in various forms for various devices and users want to use seamlessly the contents with an appropriate device. Web browser extension is function to add features that are not supported by default browser. All browsers support extensions that provide the same services for cross-browser. In this paper, We proposed object migration architecture between heterogeneous browsers by expanding our proposed mechanism that identifies objects and the information of those objects to be migrated in the web page, extracts the object and creates object after migration. For this purpose, we analyzed the extension architecture of representative browsers and investigated necessary files to develop objects migration extension. In addition, We investigated how to send and receive message among files in each browser extension and the interaction mechanism among those files. Finally, We implemented the object migration mechanisms between heterogeneous browsers.

Key words: Browser Extension(브라우저 확장기능), Object Migration(객체 이동), HTML5, Heterogeneous Browsers(이종 브라우저)

※ 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 : 2012년 10월 5일, 수정일 : 2012년 12월 9일

완료일 : 2013년 1월 4일

[†] 준회원, 동의대학교 디지털미디어공학과
(E-mail : sej1272@naver.com)

^{**} 정회원, 동의대학교 영상정보공학과

※ 본 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기본연구 지원사업(No.2010-0025069) 및 2012년도 동의대학교 교내연구비(과제번호2012AA206)의 지원을 받아 수행된 연구임.

1. 서 론

다양한 웹 서비스를 제공받을 수 있는 스마트폰, 태블릿 PC, 스마트 TV와 같은 단말이 증가함에 따라 사용자는 언제, 어디서나 웹을 접할 수 있게 되었다. 웹에는 여러 가지 콘텐츠들이 존재한다. 그 콘텐츠들은 각각 정보들을 가지고 있는데, 두 개 이상의 콘텐츠가 모여서 융합된 콘텐츠로 사용자에게 제공될 수 있다. 이에 따라 사용자는 융합 콘텐츠를 사용자가 가지고 있는 단말에 관계없이 자유롭게 이동하여 사용하려는 요구가 증가하고 있다. 그러나 단말이 종류가 각각 다르므로 디스플레이 크기, OS의 종류에 따라 콘텐츠가 제공되지 않을 수 있어, 사용자가 원하는 모양과 크기로 콘텐츠가 나타나지 않을 수 있다. 그러므로 단말의 종류나 플랫폼에 상관없이 제공되는 HTML(Hyper Text Markup Language)⁵ 기술이 요구된다. HTML5에는 기존의 HTML에서 지원하는 기술에 멀티미디어 콘텐츠(Video, Audio), Canvas 기능 등을 추가로 지원하여 웹으로 무궁무진한 콘텐츠를 제작할 수 있도록 지원하고 있다. 그리고 HTML5 기반 스트리밍 서비스의 단말 간 이동에 대한 연구도 이루어졌다[1]. 웹 페이지의 콘텐츠들은 각각을 객체로 생각할 수 있다. 여러 객체들 중 사용자가 관심 있는 객체만 골라 사용자의 단말에서 볼 수 있다면 사용자는 웹 서비스를 제공받는데 편리함을 느끼고 웹을 더 효율적으로 사용할 것이다.

브라우저 확장기능(Extension)은 웹 브라우저의 기본 기능에 포함되지 않은 기능을 추가적으로 외부 개발자가 개발하여 브라우저에 설치해 사용할 수 있도록 제공하는 기능이다. 이것은 다양한 부가기능을 제공하여 사용자가 웹을 사용하는데 불편함이 없도록 편리성을 제공하고 있다. 또한 현재 구글의 크롬, 모질라의 파이어폭스, 오페라 소프트웨어의 오페라, 애플의 사파리와 같이 전 세계적으로 널리 사용되고 있는 대표적인 웹 브라우저는 확장 프로그램을 지원하며, 자체적인 확장기능 스토어를 제공하고 있다. 사용자가 스토어에서 필요한 확장기능을 다운받으면 웹 브라우저의 툴바(Tool Bar)에 아이콘이 추가된다. 설치한 확장기능에 대해 각 브라우저의 확장기능 메뉴를 통해 사용안함, 제거, 재사용 등을 설정할 수 있다. 웹 브라우저는 또한 사용자가 직접 개발할 수 있도록 확장기능에 대한 각종 개발문서와 예제를

지원하고 있다[2-5]. 이러한 확장기능을 통해 다양한 서비스를 각 브라우저에 적합하게 제작해 배포함으로써 사용자의 브라우저에 관계없이 동일한 서비스를 제공할 수 있으며 웹 소켓(Web Socket)[6]의 등장으로 다른 브라우저 간에도 서로 상호작용이 가능하다.

본 논문에서는 웹 페이지에 존재하는 객체를 식별하고 이동을 위해 필요한 정보를 구별하는 방법과 객체를 추출하여 이동 후 생성되는 방법에 대해 수행한 우리의 연구 결과[1,7]를 이종 브라우저 간 객체 이동으로 확장하기 위한 이종 브라우저 간 객체 이동 구조를 제시하고 크롬과 사파리 브라우저를 이용하여 구현하였다. 웹브라우저는 사용자가 이용하기에는 동일한 UI(User Interface)를 제공하지만 브라우저마다 확장기능의 구조는 차이가 있다. 본 논문에서는 각 브라우저의 확장기능이 개발자에게 제공되는 요소와 요소 간 메시지 전달 메커니즘을 분석하였으며, 객체 이동 메커니즘을 이종 브라우저에 적용하여 다른 브라우저 간에도 객체 이동이 가능함을 보인다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 객체 이동과 관련 기존 기술에 대해 설명하고 3장에서는 기존에 연구한 객체 식별, 객체 정보 구별, 객체 추출과 생성 방법에 대해 간단히 설명한다. 4장에서 각 브라우저의 확장기능 개발에 필요한 파일에 대해 살펴보고 5장에서는 각 파일들이 상호작용하는 방법과 웹 페이지와 상호작용하는 방법에 대해 알아본다. 6장에서는 다른 브라우저로의 웹 객체 이동의 구현 모습을 설명한다. 마지막으로 7장에서 결론을 맺는다.

2. 관련 기술

웹을 사용하는 사용자들은 자신이 필요한 정보를 저장하기 위해 웹 페이지를 즐겨찾기에 저장해 놓거나 웹 페이지의 URL(Uniform Resource Location)을 기억하여 직접 입력해서 이용한다. 그러나 웹 페이지에 자신이 필요한 정보가 부분적으로 포함되어 있거나 웹 페이지의 URL을 기억하지 못한다면 다시 검색해서 찾아야 한다. 이를 위해 웹 페이지의 일부를 저장하는 서비스가 존재한다. 각 서비스는 객체 선택, 객체 저장, 저장된 객체를 사용자에게 제공하는 형태에 따라 차별화된 서비스를 제공하고 있다.

Wapware[8]는 크롬의 확장기능으로 제공되며, 위젯 형태로 사용자가 회원가입을 통해 계정을 만들어서 사용자가 저장한 정보가 모두 웹서버에 저장된다. 사용자가 관심 있는 영역을 마우스로 드래그 하면 영역을 선택하고 저장할 수 있다. 또한 개인 페이지를 제공하여 사용자가 저장한 객체가 각각의 레이아웃으로 설정되어 이용할 수 있다. 그러나 선택된 영역을 보여줄 때 웹 페이지 전체를 로드하고 화면에는 사용자가 선택한 부분만 보여준다. 따라서 네트워크 자원의 낭비가 발생한다.

Memonic[9]은 익스플로러, 파이어폭스, 크롬, 사파리, 오페라에서 확장기능으로 제공되며, Wapware와 마찬가지로 사용자가 계정을 만들고 저장한 정보는 웹서버에 저장된다. Memonic은 웹 페이지 내에 있는 객체를 그룹화하여 저장한다. 그리고 웹 페이지 전체를 저장하거나 북마크를 설정할 수 있다. Memonic도 개인 페이지를 제공하여 사용자가 저장한 객체들을 모두 볼 수 있다. Web slice[10]는 익스플로러 8 이상에서 동작되며, 사용자가 원하는 웹 사이트의 특정 부분만 잘라내어 실시간으로 업데이트되는 내용을 확인할 수 있다. 또한 어떤 정보든 전부 제공받을 수 없고 웹 페이지에서 이 기능을 지원해주는 부분만 사용가능하다. 콘텐츠 제공자가 서비스를 제공하고자 하는 부분에 <div> 태그와 'hslice'라는 클래스 속성 값을 이용하여 생성한다.

File clip[11]은 파이어폭스에서 제공된다. 객체 위에 마우스 오버를 하면 사용자가 필요한 객체를 선택하여 저장하며 저장된 객체는 개인 페이지에서 확인할 수 있다. Web chunk[12]는 파이어폭스에서 제공되는데, 익스플로러의 Web slice 서비스를 파이어 폭스에서 사용하기 위한 확장기능이다. Arbitrary web slice[13]는 크롬에서 제공되는 확장기능이며, Web slice와 기능이 유사하다. 이것은 콘텐츠 제공자가 서비스를 제공할 부분에 <div> 태그의 id를 이용해 설정하고, 확장기능을 실행하면 확장기능이 웹 페이지에 있는 <div> 태그 중 id를 검색하여 그 부분을 팝업 페이지에 나타낸다.

Pinterest[14]는 트위터(Twitter)나 페이스북(Facebook)과 같은 SNS(Social Network Service)로, 사용자가 관심있는 정보를 보드판에 핀을 꽂듯이 게시하여 다른 사용자들과 공유한다. 정보를 추가할 때는 이미지 또는 동영상의 URL을 입력하거나 컴퓨

터에서 파일을 업로드할 수 있다. 보드 이름, 카테고리, 핀보드에 업로드가 가능한 사람 및 보드판을 설정할 수 있고, 다른 사용자가 올린 핀을 자신의 핀보드로 가져와 공유할 수 있다.

File clip, Web chunk, Arbitrary Web Slice는 현재 웹브라우저의 최신버전에서 실행되지 않고 있다. Wapware와 Memonic, Pinterest는 개인 페이지에서 다른 사용자들에게 공유가 가능하지만 Web slice는 불가능하다. 그리고 Wapware는 객체를 그룹화하여 선택할 수 없고, 또한 이것은 Memonic과 더불어 각각 저장한 객체들을 결합할 수 없다. Pinterest는 두 경우 모두 지원되지 않는다. 따라서 이러한 단점들을 보완하여 객체 선택 시 그룹화 된 객체를 한 번에 선택하거나 저장한 객체들을 결합하고 다른 브라우저를 사용하는 다른 사용자에게 바로 전달하는 방법이 요구된다.

3. 웹 객체 이동 구조

웹 객체 이동을 위해서는 객체를 식별하고 이동에 필요한 정보를 알아야 한다. 그리고 그 정보들을 바탕으로 객체를 추출하고 이동 후 객체를 생성하는 방법을 고려해야 한다. 웹 객체의 정보를 이동하기 위해서 단말 간에는 직접적으로 메시지를 주고받을 수 없기 때문에 그림 1과 같이 서버를 이용한다. 메시지를 주거나 받기 위해서는 단말이 서버에 접속되어 있어야 하고, 서버가 사용자를 관리하여 다른 사용자에게 전달한다[7].

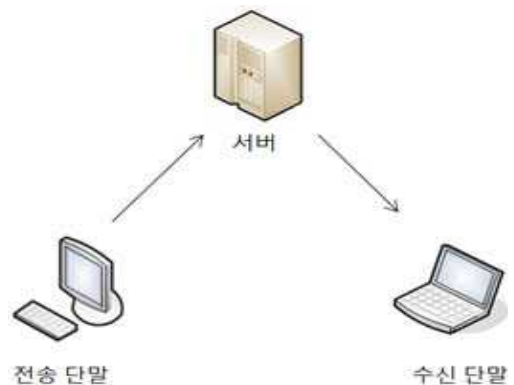


그림 1. 웹 객체 전송을 위한 기본 구조

3.1 객체 식별 및 이동 정보

웹 객체에는 각각 태그(tag) 이름이 존재한다. 각

태그는 여러 속성들을 가지는데 그 중 공통적인 속성으로 class, id, style, title 등이 존재한다. id 속성은 객체를 구별할 수 있고 각 객체마다 유일한 값을 가진다. class(이하 클래스 이름이라 사용) 속성은 하나의 객체에 설정하거나 여러 개의 객체들에게 공통적인 속성 값을 설정하여 CSS(Cascading Style Sheets)를 적용하는데 사용한다.

그리고 식별한 객체를 이동하여 다시 생성하기 위해 필요한 객체의 정보 추출이 필요하다. 전달될 정보는 객체의 특성에 따라 URL 정보, value 속성 값, innerHTML 속성 값으로 특징지어진다. 객체를 이동하기 위해서는 이들 정보를 각각 추출하는 것이 필요하다.

3.2 객체 추출 및 생성

웹 페이지는 독립적인 객체 또는 서로 관련성이 높은 객체로 구성된다. 본 논문에서는 웹 페이지 제공자가 관련이 있다고 생각하는 객체를 같은 클래스 이름을 사용하여 그룹으로 구성해 놓고 사용자가 그 객체 중 하나를 선택하면 관련이 있는 객체들이 모두 선택되는 방법과 그룹화 된 객체 외에 나머지 객체들 중 사용자가 함께 이동하려는 객체가 있는 경우 함께 선택하여 이동하는 방법[7]을 적용한다.

선택한 객체들을 이동하기 위해서는 객체들의 정보를 정확하게 추출하여 각각의 정보를 메시지로 패키징해야 한다. 객체들의 정보는 객체의 특성에 맞게 URL, value, innerHTML을 기준으로 구분하여 객체의 태그 이름과 부가 정보를 함께 JSON[15] 형태로 저장하여 전달한다.

전달받은 객체를 생성하기 위해서는 객체의 이름을 사용하여 document.createElement() 메소드를 호출한다. 객체를 생성하고 주요 정보와 부가 정보를 이용해 속성값을 설정한 후 document.body.appendChild() 메소드를 호출하여 새로 생성된 객체를 웹 문서의 <body> 영역에 추가하여 웹 객체를 생성한다.

4. 브라우저 확장기능에 필요한 파일

브라우저의 확장기능을 개발하기 위해 여러 파일들이 사용되며 브라우저 마다 정의하고 있는 파일과 의미와 기능이 차이가 있다. 확장기능을 이용하여 객

체를 이동하기 위해서는 객체를 이동하는데 필요한 기능이 어떤 파일에 적용되어야 하는지 알아야 한다. 각 브라우저의 확장기능 개발에 필요한 파일의 종류와 기능은 크롬과 사파리로 나누어 정리한다.

4.1 구글 크롬

크롬에서 확장기능 추가에 사용하는 파일은 매니페스트 파일, HTML 파일, 자바스크립트 파일, 이미지 파일이다. 매니페스트 파일은 JSON 형식으로 확장기능에 대한 정보를 포함한다. 여기에는 확장기능의 이름, 버전, 사용하는 아이콘, 백그라운드 페이지 이름 등이 기술된다. 파일명을 manifest.json이고 파일명을 변경할 수 없다. 또한 이 파일은 확장기능을 제공할 때 필수적으로 생성한다.

HTML 파일은 백그라운드와 UI(User Interface) 페이지를 위해 하나 이상 필요하다. 백그라운드 페이지는 사용자에게 보이지 않고 확장 프로그램 실행 관련 주 알고리즘이 기술되며, 확장기능에서 하나만 생성된다. 이것은 하나의 브라우저에서 여러 개의 탭이 존재할 때 모든 탭에 동일하게 처리가 필요할 때 사용한다. UI 페이지는 확장기능의 UI를 나타내는 HTML 파일이다. 이 페이지에는 팝업이나 새로 생성된 HTML 파일을 포함한다. 팝업은 확장기능 아이콘이 눌렸을 때 실행되는 팝업창을 띄우는데 사용한다. 새로 생성된 HTML 파일은 chrome.tabs.create() 메소드 또는 window.open() 메소드를 사용하여 새로운 HTML 페이지를 생성한다.

자바스크립트 파일은 브라우저에서 로드된 웹 페이지의 콘텐츠를 제어하기 때문에 콘텐츠 스크립트(Content Script) 파일이라고도 한다. 이것은 각각의 페이지마다 적용되어 페이지마다 독립적으로 실행시키고 싶은 동작이 있는 경우 사용한다. 그리고 그 외에 아이콘 이미지, 다른 자바스크립트 파일 등을 필요에 따라 생성하여 사용한다.

크롬의 확장기능 구조[16]는 그림 2와 같다. 웹 페이지에서 실행한 동작을 콘텐츠 스크립트에서 캡처하여 Parent Extension으로 보내준다. Parent Extension에는 백그라운드 페이지와 팝업 페이지 외 여러 페이지가 존재한다. Parent Extension에서 웹 페이지에 적용할 동작은 콘텐츠 스크립트에게 메시지를 보내 상호작용한다. 그리고 팝업 페이지와 백그라운드 페이지도 상호작용이 가능하다.

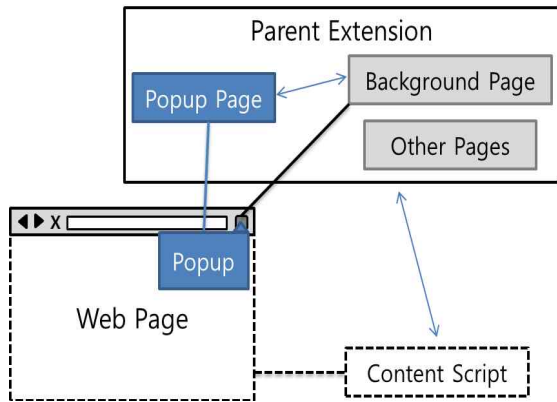


그림 2. 크롬 확장기능 구조

4.2 애플 사파리

사파리 확장기능 사용에 필요한 파일은 글로벌 HTML 페이지, 콘텐츠 파일, 스크립트 파일(Injected scripts), 스타일 시트 파일(Injected Style Sheets), 아이콘, 이미지나 미디어 파일이 필요하다.

글로벌 HTML 페이지는 보이지 않지만 HTML 요소들을 포함하고 있으며 확장기능을 설치하거나 활성화할 때 한번만 로드된다. 이 파일도 크롬의 백그라운드 페이지처럼 웹 페이지 전체에 대한 동작을 실행시키고 싶을 때 사용한다. 콘텐츠 파일은 확장바(Extension Bar), 팝업, full-page tabs를 나타내는 HTML, CSS, 자바스크립트 파일을 포함한다. 스크립트 파일은 자바스크립트 파일로써, 브라우저에 있는 콘텐츠 파일을 제어할 수 있으므로 각각의 웹 페이지마다 적용할 기능을 스크립트 파일에 기술한다. 스타일 시트 파일로 스타일을 추가/수정할 수 있으며 아이콘은 아이콘 이미지 파일을 나타내고 그 외의 다른 사진이나 미디어를 추가할 수 있다.

사파리 확장기능 구조[17]는 그림 3과 같다. 사파리 확장기능은 웹 페이지 콘텐츠(Webpage Content)와 상호작용하는 부분과 사파리 애플리케이션(Safari Application)과 상호작용하는 부분으로 나뉜다. 웹 페이지 콘텐츠는 웹 페이지에 있는 콘텐츠를 나타내고 사파리 애플리케이션은 브라우저에 있는 메뉴, 윈도우, 탭, 툴바, 팝업을 가리킨다. 웹 페이지 콘텐츠는 자바스크립트와 CSS를 통해 상호작용하고, 사파리 애플리케이션은 글로벌 HTML 페이지, 확장 바 페이지, 팝업 페이지를 통해 상호작용한다. 사파리 애플리케이션이 웹 페이지 콘텐츠에 어떤 동작을 실행하고 싶으면 웹 페이지 프록시(proxy)를

통해 자바스크립트에게 메시지를 보내고, 자바스크립트는 그 동작을 웹 페이지 콘텐츠에 실행하게 된다. 이와 유사하게 자바스크립트도 탭 프록시를 통해 사파리 애플리케이션에 메시지를 보낸다.

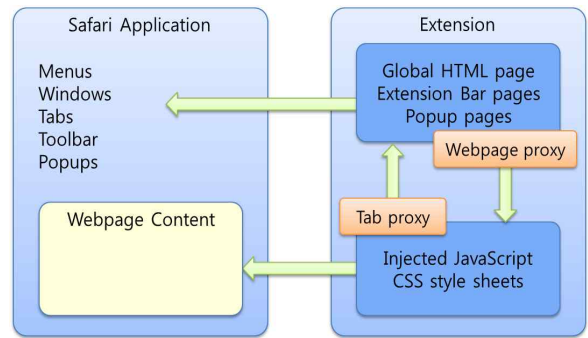


그림 3. 사파리의 확장기능 구조

5. 확장기능 별 페이지 상호작용 구조

브라우저마다 확장기능을 설치한 후 백그라운드 프로세스, 팝업, 스크립트가 서로 상호작용을 한다. 확장기능의 파일은 브라우저의 웹 페이지와도 상호작용을 할 수 있다. 다른 브라우저 간 웹 객체 이동을 위해서는 객체들의 정보를 웹 페이지와 확장기능이 서로 주고받으면서 전달해야 하기 때문에 상호작용 구조에 대해 살펴본다.

5.1 구글 크롬

크롬의 페이지 상호작용 구조[18]는 그림 4와 같다. 우선 콘텐츠 스크립트에서 connect() 메소드를 호출하여 팝업과 연결을 요청하면 팝업이 onConnect 이벤트를 통해 콘텐츠 스크립트와 연결하고, 콘텐츠 스크립트가 postMessage() 메소드를 호출하여 팝업으로 메시지를 전달하면 onMessage 이벤트로 메시지를 수신한다. 팝업에서 콘텐츠 스크립트로 메시지를 보내기 위해서는 connect() 메소드 없이 postMessage() 메소드를 호출하여 사용하고 콘텐츠 스크립트가 메시지를 받을 때는 onMessage 이벤트를 사용한다(그림 4 참조 ①).

팝업과 백그라운드 페이지가 상호작용을 하기 위해서는 getBackgroundPage() 메소드를 호출한다. 팝업에서 getBackgroundPage() 메소드를 호출하면 백그라운드 페이지의 window객체를 반환받게 된다.

그래서 백그라운드 페이지에 포함되어 있는 함수와 변수를 팝업에서 사용할 수 있다(②).

콘텐츠 스크립트와 백그라운드 페이지간 메시지 교환은 sendMessage() 메소드를 사용하나 콘텐츠 스크립트는 chrome.extension의 sendMessage() 메소드를 백그라운드 페이지는 chrome.tabs의 sendMessage() 메소드를 사용한다. chrome.extension은 모든 확장기능 페이지에서 사용할 수 있는 유틸리티로, 확장기능 및 콘텐츠 스크립트 간 메시지 교환을 지원한다. chrome.tabs는 브라우저의 탭과 상호작용을 하는 모듈로써 탭을 생성하거나 수정, 정렬을 할 수 있다. sendMessage() 메소드를 호출하면 지정한 탭에 메시지를 보낼 수 있다. 메시지를 받을 때는 둘다 같은 방식으로 onMessage 이벤트를 통해 수신한다(③).

크롬의 확장기능에서 사용하는 파일들과 새로 생성된 웹 페이지 간 상호작용 구조는 다음과 같다. 백그라운드 페이지 또는 팝업에서 chrome.tabs.create() 메소드 또는 window.open() 메소드로 새 탭이나 새 창을 생성한다(④). 생성된 웹 페이지가 백그라운드 페이지와 상호작용하기 위해서는 팝업과 백그라운드 페이지가 상호작용하는 방법과 동일하게 getBackgroundpage() 메소드를 사용한다(⑤). 또한 웹 페이지가 팝업과 통신하기 위해 웹 페이지에서 getViews() 메소드를 사용하면 확장 기능이 웹 페이지로 팝업의 window 객체를 배열로 반환한다(⑥). 팝업은 확장기능에서 여러 개가 존재할 수 있으므로 각각의 window 객체가 반환된다. getViews() 메소드의 매개변수로 int형의 windowId 값을 전달하면

해당 팝업의 window객체를 반환 받는다. 웹 페이지는 반환 받은 팝업의 window 객체를 통해 팝업과 상호작용할 수 있다. 그리고 콘텐츠 스크립트는 기본적으로 웹 페이지와 상호작용이 가능하다(⑦).

5.2 애플 사파리

먼저 확장 바 또는 팝업, 글로벌 HTML이 스크립트와 메시지를 주고받는 방법은 그림 5와 같다[19]. 확장 바 또는 팝업, 글로벌 HTML과 스크립트 파일은 서로에게 메시지를 전달할 때 dispatchMessage() 메소드를 호출한다. 그리고 메시지를 전달 받기 위해서는 addEventListener() 메소드를 호출하여 'message' 이벤트의 핸들러를 등록한다. 각 파일에서 메시지를 전달받으면 message 이벤트가 발생하여 사용자가 정의한 함수가 실행된다.

확장 바 또는 팝업이 글로벌 HTML과 상호작용하는 방법은 그림 6과 같다[20]. 확장 바 또는 팝업에서 safari.extension.globalPage의 contentWindow 속성을 사용하면 글로벌 HTML의 window객체를 반환해 준다. 그 후 글로벌 HTML에 있는 함수와 변수를 확장 바 또는 팝업에서 사용한다.

확장기능이 새로 생성된 웹 페이지와 상호작용하는 방법은 그림 7과 같다[21]. 확장 바 또는 팝업, 글로벌 HTML에서 openTab() 메소드 또는 openBrowserWindow() 메소드를 호출하면 새로운 탭이나 창이 생성된다. 웹 페이지는 활성화가 되어야 스크립트 파일과 상호작용을 할 수 있다. 그러므로 확장 바 또는 팝업, 글로벌 HTML에서 웹 페이지를 생성하면 활성화 시켜준다. 확장 바 또는 팝업, 글로

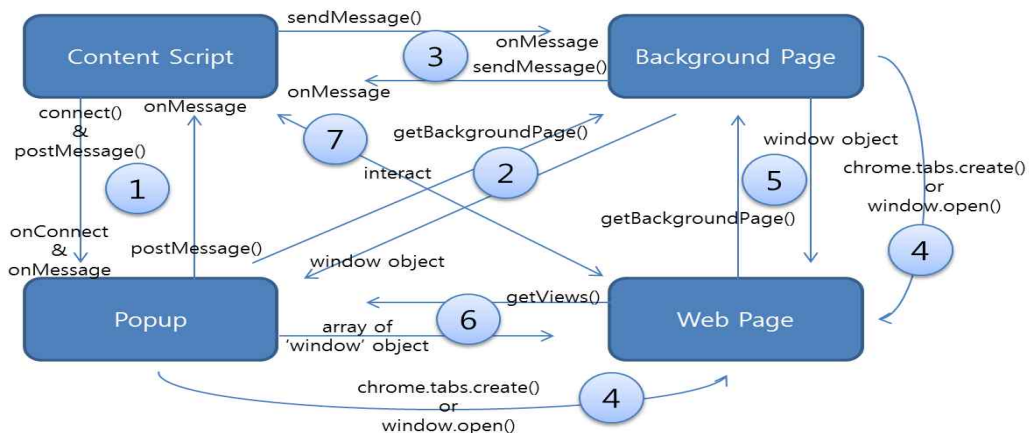


그림 4. 크롬 확장기능 파일 간 상호작용 구조

별 HTML는 직접적으로 웹 페이지와 상호작용할 수 없으므로 스크립트 파일을 통해 웹 페이지와 상호작용이 가능하다.

크롬과 사파리의 확장기능 내 파일 간 메시지 전달구조를 살펴보면, 크롬에서 콘텐츠 스크립트와 웹 페이지는 chrome.tabs로, 팝업과 백그라운드 페이지는 chrome.extension으로 각각 메시지를 전송할 수 있다. 웹 페이지와 백그라운드 페이지는 메시지 전송이 아닌 웹 페이지에서 백그라운드 페이지의 윈도우 객체를 반환받아 속성 설정 또는 메소드 호출을 통해 정보를 전달한다. 사파리는 사파리 어플리케이션 부분과 웹 페이지 콘텐츠 부분으로 나뉘며, 두 부분이 직접적으로 메시지를 전송할 수 없기 때문에 웹페이지는 스크립트 파일을 통해 통신이 가능하다. 이처럼 브라우저 마다 확장 기능을 구성하는 파일 간 메시지 전달구조가 다르기 때문에 이동 구조 설계 시 반영하여야 한다.

6. 확장기능 별 객체 이동 및 생성 절차

웹 이용 중 웹 페이지에 사용자가 관심 있는 콘텐츠를 다른 단말의 브라우저로 이동하여 단말의 크기에 맞게 보고 싶거나, 다른 사용자와 공유하고 싶은

경우가 있다. 이 장에서는 확장기능을 이용하여 다른 브라우저 간 객체를 이동하는 절차를 알아본다. 이 절차는 확장기능을 설치하고 사용자가 웹을 이용하는 도중 이동하고 싶은 객체를 발견하고 확장기능을 활성화한 뒤 실행하는 모습이다.

6.1 구글 크롬

크롬에서 객체를 이동하기 위해서는 콘텐츠 스크립트, 백그라운드 페이지, 팝업, 객체를 생성할 HTML 페이지가 필요하다. 객체를 추출하고 메시지화 하여 서버로 전달하는 모습은 그림 8과 같다.

먼저 사용자는 팝업에서 자신의 아이디, 비밀번호로 로그인을 하면(그림 8 참조 ①) 서버에서 사용자 인증을 한다(②). 사용자가 객체를 클릭하면 콘텐츠 스크립트에서는 선택한 객체들을 표시하고, 객체 추출 시 다른 객체들과 구별하기 위해 그룹화 한다(③). 관심 있는 객체가 모두 선택되면 팝업을 클릭하여 활성화시킨다. 팝업이 활성화되면 콘텐츠 스크립트는 그룹핑된 객체들을 팝업으로 전달한다(④). 팝업에서는 콘텐츠 스크립트에서 전달받은 메시지를 전달할 사용자를 선택하여 전달을 누르면 객체와 전달할 사용자가 메시지화되어 백그라운드 페이지로 전달된다(⑤). 백그라운드 페이지는 서버와 연결(⑥)하

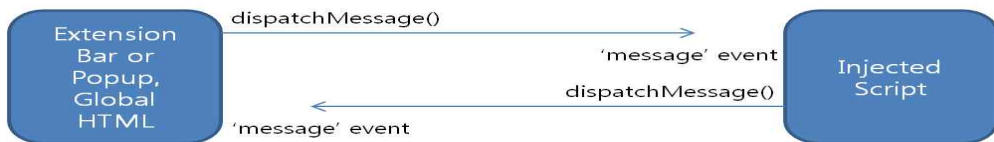


그림 5. 확장 바 또는 팝업, 글로벌 HTML과 스크립트 파일 간 메시지 통신 방법



그림 6. 확장 바 또는 팝업과 글로벌 HTML 간 상호작용



그림 7. 사파리 확장기능과 웹 페이지 간 상호작용

여 서버를 통해 다른 브라우저에게 객체 정보를 메시지화 하여 전달한다(⑦).

서버로부터 객체 정보가 전달되어 새로운 웹 브라우저에서 객체가 생성되는 절차는 그림 9와 같다. 먼저 팝업에서 아이디와 비밀번호를 입력하여 로그인을 하면(그림 9 참조 ①) 백그라운드 페이지에서 그 정보들을 서버로 보내 사용자 인증을 한다(②). 인증을 한 뒤, 만약 전달받은 객체 정보가 있으면(③) 백그라운드 페이지는 알림 메시지를 띄워준다(④). 전달받은 메시지를 확인하기 위해 팝업에서 새 탭 생성 버튼을 누른다(⑤). 팝업에서 버튼을 누르면 새로운 웹 페이지(Other Page)가 활성화 된다. 새로운 웹 페이지는 서버로부터 전달받은 메시지를 백그라운드 페이지로부터 가져와 새로운 웹페이지 창에 객체를 생성해 주는 역할을 한다(⑥). ⑥에서, 서버로부터 전달받은 메시지를 백그라운드 페이지에서 콘텐

츠 스크립트로 전달하여 객체를 생성하는 방법도 있다. 그러나 그 방법은 백그라운드 페이지가 서버로부터 받은 메시지를 콘텐츠 스크립트로 보내고 이를 다시 웹 페이지로 전달하여야 하고 웹 페이지가 미리 활성화되어있어야 한다. 따라서 웹페이지가 활성화될 때 백그라운드 페이지의 윈도우 객체를 이용하여 메시지를 받아 객체를 생성하는 방법을 선택하였다.

6.2 애플 사파리

사파리에서 객체 이동을 위해 필요한 파일은 스크립트 파일(Injected Script), 확장 바, 글로벌 HTML가 필요하다. 그리고 사파리에서는 팝업대신 확장 바를 사용한다. 확장 바는 아이콘 없이 툴 바(Tool Bar)처럼 웹 페이지의 메뉴 밑에 길게 붙어있는 도구이다. 파일들의 기능은 각각 크롬의 콘텐츠 스크립트, 팝업, 백그라운드 페이지와 같다.

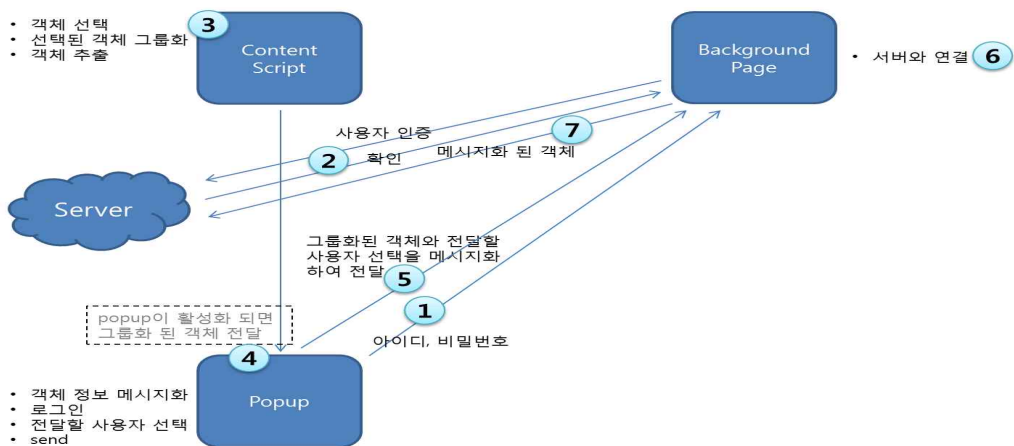


그림 8. 크롬 확장기능의 객체 추출 및 전달

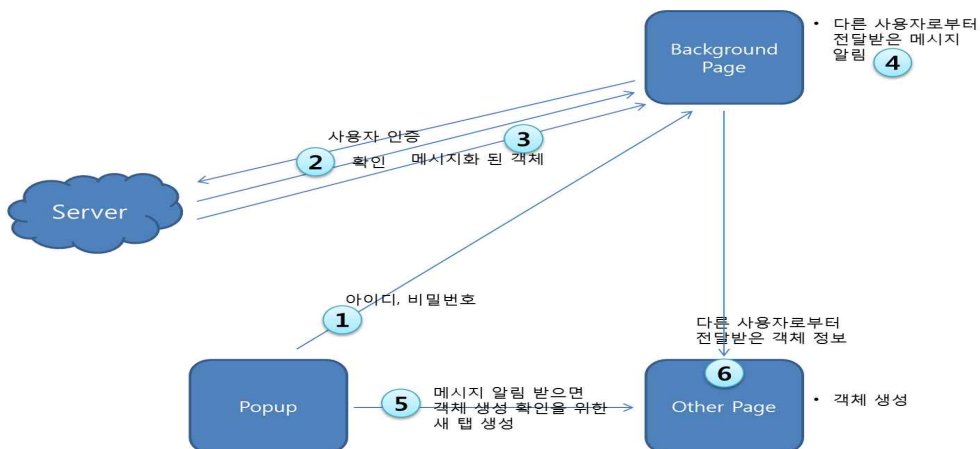


그림 9. 크롬 확장기능의 객체 수신 및 생성

먼저 사파리 확장기능의 객체 추출 및 전달 과정은 그림 10과 같다. 확장 바에서 사용자의 아이디, 비밀번호를 입력하면(그림 10 참조 ①) 서버에서 사용자 인증을 한다(②). 사용자가 객체들을 선택하면 스크립트 파일에서 선택한 객체들을 표시하고 그룹화 한다(③). 객체 선택이 끝나면 import 버튼을 통해(④) 확장 바로 객체들이 전달된다(⑤). import 버튼을 사용하는 이유는 확장 바가 스크립트 파일로부터 객체 정보를 전달받기 위해서는 이벤트를 통하여 메시지로 전달받아야 하기 때문이다. 그리고 확장 바에서 전달할 사용자 정보와 객체 정보를 메시지화 한 뒤 글로벌 HTML로 전달한다(⑥). 글로벌 HTML은 서버와 연결하여(⑦) 서버로 메시지를 전달한다(⑧).

서버로부터 객체 정보를 수신하고 생성하는 절차는 그림 11과 같다. 먼저 아이디와 비밀번호를 통해 로그인을 한다(그림 11 참조 ①). 서버에서 사용자

인증을 하고(②) 전달받은 메시지가 있으면(③) 글로벌 HTML은 알림 메시지를 띄워준다(④). 사용자는 확장 바에서 새 탭 생성 버튼을 클릭한다(⑤). 새 탭 생성 버튼을 클릭하면 새로운 웹 페이지(Other Page)가 생성되는데, 새로운 웹 페이지가 활성화되면(⑥) 확장 바는 글로벌 HTML에게 Other Page가 생성됨을 알린다(⑦). 글로벌 HTML은 알림을 받은 후 서버에서 전달받은 객체를 다시 스크립트 파일로 전달한다(⑧). 스크립트 파일에서는 메시지화된 객체 정보를 파싱(Parsing)하여 생성한 뒤(⑨) 새로운 웹 페이지의 <body> 태그에 덧붙인다(⑩). 새 탭 생성 시 글로벌 HTML이 새로운 웹 페이지로 객체를 바로 줄 수 없는 이유는 일반 웹 페이지와 확장기능 파일 간 주고받는 메시지 형식이 사파리에서는 제공되지 않고, 사파리 구조에서 본 것과 같이 웹페이지 콘텐츠와 상호작용하는 것은 스크립트 파일이기 때

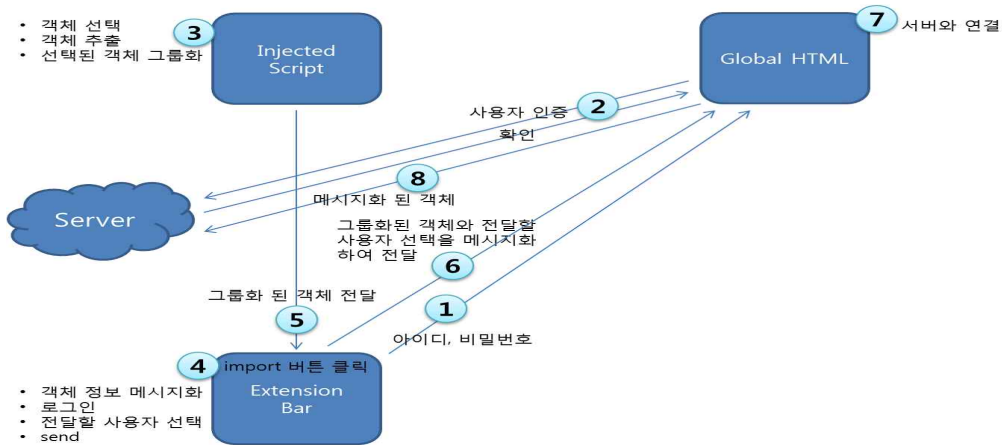


그림 10. 사파리 확장기능의 객체 추출 및 전달

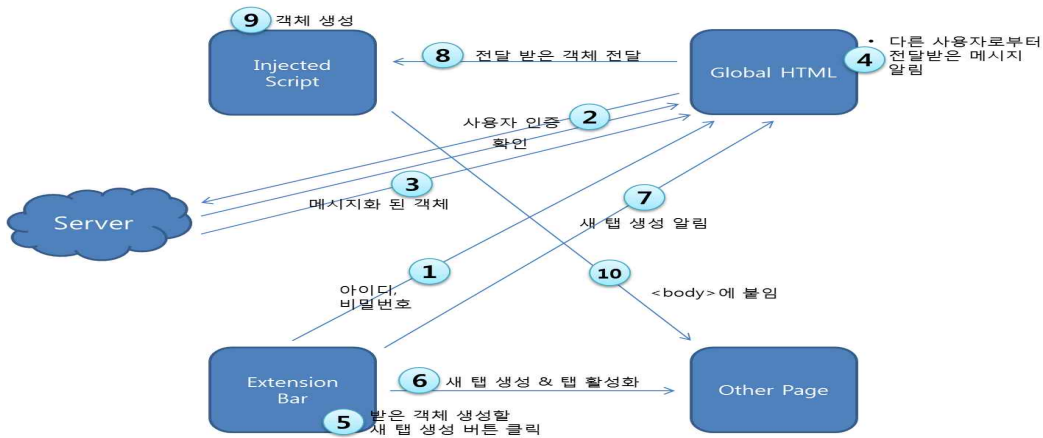
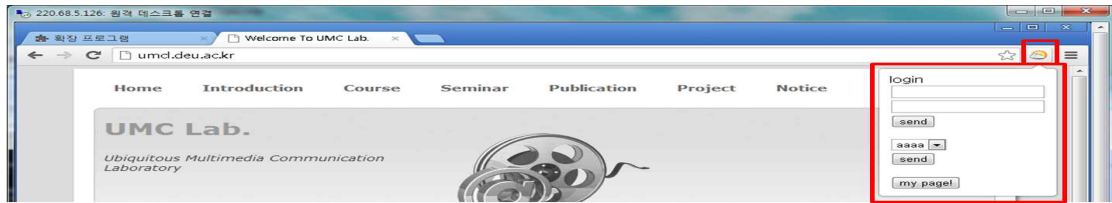


그림 11. 사파리 확장기능의 객체 수신 및 생성

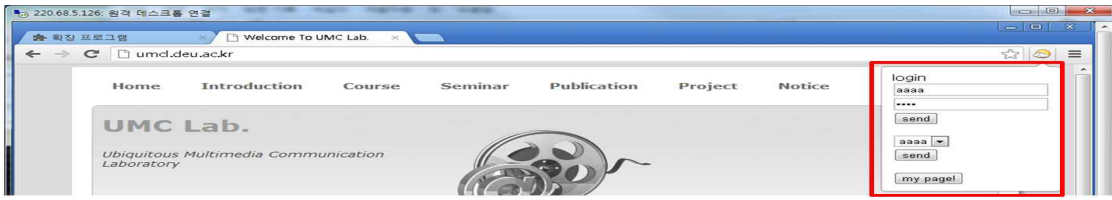


(a)



(b)

그림 12. 확장기능 설치 모습

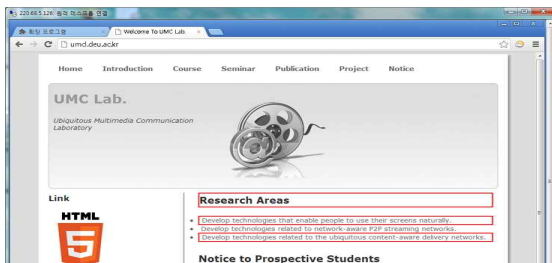


(a)

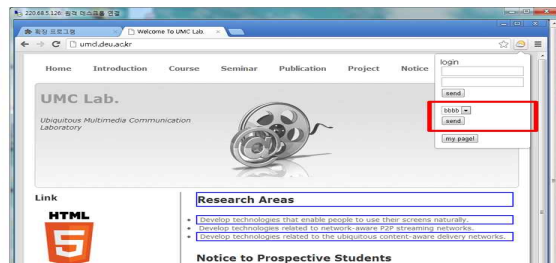


(b)

그림 13. 아이디와 비밀번호 입력

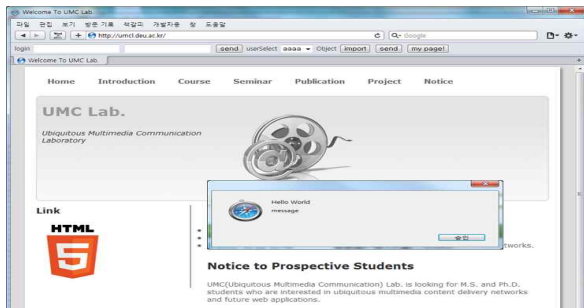


(a)



(b)

그림 14. 객체 선택, 전달할 사용자 선택



(a)



(b)

그림 15. 객체 수신, 객체 생성 모습

문이다.

사파리의 객체 이동 구조를 설계하고 크롬의 객체 이동 구조와 비교하여 이중 브라우저 간 객체를 이동한 결과 화면은 그림 12-그림 15와 같다.

서버와 통신하기 위해 다른 두 대의 PC를 사용하여 한 PC의 브라우저에는 크롬 브라우저를 사용하고 다른 PC에는 사파리 브라우저를 사용한다. 먼저 확장기능을 설치한 모습은 그림 12와 같다. 그림 12의 (a)는 크롬에서 팝업의 형태로 구현한 모습이고 그림 12의 (b)는 사파리에서 확장 바의 형태로 구현한 모습이다. 두 브라우저 모두 로그인을 할 수 있는 창이 있고 객체를 전달하려는 사용자를 선택할 수 있는 옵션창이 있다. 또한 전달받은 객체를 새로운 웹 페이지에서 볼 수 있는 버튼이 있다. 그리고 사파리는 선택한 객체를 확장기능으로 불러올 수 있는 import 버튼이 존재한다.

이렇게 설치를 한 뒤 서버와 연결하고 그림 13과 같이 서로 다른 아이디와 비밀번호를 로그인 창에 입력한다. 각각 send 버튼을 누르면 서버에 아이디, 비밀번호가 전달되어 로그인이 된다.

그 이후 객체를 선택하면 그림 14의 (a)와 같이 테두리로 표시가 된다. 사용자가 전달하고 싶은 객체를 모두 클릭한다. 선택을 취소하고 싶으면 객체를 다시 클릭한다. 이렇게 선택한 객체들은 동일한 class 속성 값을 가지고 그룹화가 된다. 사용자가 모두 선택한 뒤 팝업을 클릭하면 위 그림 14의 (b)에서는 확인할 수 없지만 실제로는 테두리 색이 바뀌게 되고, 선택한 객체들의 정보들이 메시지화 되어 팝업으로 전달된다. 그리고 사용자는 전달할 사용자를 선택하고 보내기 버튼을 누른다. 객체들은 서버로 전달된다.

전송한 객체가 서버를 통해 다른 PC로 전달되면 그림 15의 (a)와 같이 알림 메시지가 뜬다. 전달받기 전에 사파리는 크롬에서 선택한 전달받을 사용자의 아이디와 비밀번호로 로그인을 한다. 전달받은 객체를 확장기능의 스크립트 파일은 태그 이름으로 구분

하여 객체들을 나눈다. 그리고 태그 이름으로 객체를 생성하고 속성 값을 설정한다. 사용자가 확인 버튼을 누르고 새 탭 생성 버튼을 클릭하면 그림 15의 (b)와 같이 새 탭이 생성된다. 이때 스크립트 파일은 이 새 탭 페이지의 <body>태그에 생성한 객체들을 덧붙인다. 그리고 사용자는 객체들이 생성된 모습을 확인할 수 있다.

크롬과 사파리 브라우저의 객체 이동에 대해 공통점과 차이점은 표 1과 같다. 객체를 선택하여 서버로 보내는 것까지는 공통적이지만 새 페이지에서 객체를 생성하는 방법이 차이가 발생한다.

7. 결 론

본 논문에서는 기존에 연구한 객체 이동 구조를 이용하여 서로 다른 브라우저 간 객체 이동 구조를 제안했다. 이를 위해 먼저 확장 기능 내 페이지 간 메시지 전달 구조를 분석하여 크롬 확장기능과 사파리 확장기능의 메시지 전달구조의 차이점을 발견했다. 메시지 전달 구조를 통해 사파리에서 객체 이동 구조를 설계하고, 크롬의 객체 이동 구조와 비교를 했다. 그래서 사파리에서 객체 이동 확장기능을 구현해서 이기종 브라우저 간 객체 전송이 됨을 검증했다. 본 논문에서 검증한 방법은 동일한 확장기능을 제공함으로써 사용자가 사용하는 브라우저에 관계없이 동일한 UI/UX를 경험할 수 있다. 크롬과 사파리 외에 다른 브라우저들도 이런 구조를 이용하여 브라우저가 제공하는 방식에 맞게 개발하면 어느 브라우저에 상관없이 객체를 주고받을 수 있다.

향후 사용자가 이동시킨 객체를 전달받은 브라우저에서 생성할 때 원하는 모양으로 배치할 수 있는 방법에 대해 연구하고, 다른 사람들과도 공유가 가능하게 하여 여러 사람들이 함께 사용할 수 있는 공유 스크린을 만들어 스마트폰으로 객체를 제어할 수 있는 방법에 대해 연구를 진행할 계획이다.

표 1. 객체 이동을 위해 공통점과 차이점

공 통 점	차 이 점	
객체 선택, 추출 및 그룹화, 로그인 및 전달할 사용자 선택, 백그라운드 페이지로 메시지 전달, 서버로 객체 이동, 객체 생성 페이지 존재	크롬	받은 객체를 웹 페이지에서 생성
	사파리	받은 객체를 스크립트 파일에서 생성한 뒤 웹 페이지에 덧붙임

참 고 문 헌

[1] 최헌희, 김근형, “HTML5 기반 HTTP 스트리밍 환경에서의 서비스 이동성 연구,” 멀티미디어학회논문지, Vol. 14, No. 7, pp. 905-016, 2011.

[2] Google Chrome Extensions, <http://developer.chrome.com/extensions/index.html>, 2011년 12월.

[3] Safari Extension Development Guide, <http://developer.apple.com/library/safari/documentation/Tools/Conceptual/SafariExtensionGuide/SafariExtensionGuide.pdf>, 2012년 5월.

[4] Mozilla add-on Developer Hub, <https://addons.mozilla.org/en-US/developers/>, 2012.

[5] Opera extensions: documentation, <http://dev.opera.com/addons/extensions/>, 2012.

[6] Web Socket, <http://dev.w3.org/html5/web-sockets/>, 2011년 12월.

[7] 최헌희, 송은지, 김근형, 김화숙, 조기성, “HTML5기반 융합 서비스의 서비스 이동 메커니즘 설계,” 멀티미디어학회논문지, Vol. 15, No. 4, pp. 540-551, 2012.

[8] Wepware, <http://www.wepware.com/web/main.do>, 2012.

[9] Memonic, <http://www.memonic.com>, 2012.

[10] Wep slice, [http://msdn.microsoft.com/en-us/library/cc956158\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/cc956158(v=vs.85).aspx), 2012.

[11] File clip, <http://fireclip.awardspace.info/>, 2012.

[12] Web chunk, <http://www.wepware.com/web/main.do>, 2012.

[13] Arbitrary web slices, <http://code.cheesydesign.com/?p=450>, 2012.

[14] Pinterest, <http://pinterest.com/>, 2012.

[15] JSON website, www.json.org, 2011.

[16] 크롬 확장기능 구조, <http://developer.chrome.com/extensions/overview.html>, 2011년 12월.

[17] Safari Extension Development Guide, “사파리 확장기능 구조,” pp. 15, 2012.

[18] 크롬 상호작용 구조, <http://developer.chrome.com/extensions/extension.html>, 2011.

[19] Safari Extension Development Guide, “사파리 확장 바 또는 팝업과 글로벌 HTML과 스크립트 파일 간 메시지 통신 방법,” pp. 87-90, 2012.

[20] Safari Extension Development Guide, “사파리 확장 바 또는 팝업과 글로벌 HTML 간 상호작용,” pp. 46-49, 2012.

[21] Safari Extension Development Guide, “사파리 확장기능과 웹 페이지 간 상호작용,” pp. 80-82, 2012.



송 은 지

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



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