위젯 서비스를 위한 오픈 API 프록시 서버 시스템

(An Open API Proxy Server System for Widget Services)

안 병 현 † 이 혁 준 †† 최 용 훈 ††† 정 영 욱 ††††
(Byunghyun Ahn) (Hyukjoon Lee) (Yonghoon Choi) (Younguk Chung)

요 약 위젯은 자주 이용하는 서비스를 독립적으로 구동시키는 작은 응용프로그램으로서, 콘텐츠를 제공하는 웹 사이트 등을 직접 방문하지 않고도 사용자가 원하는 콘텐츠를 손쉽게 얻을 수 있어서 많은 관심을 끌고 있다. 이러한 위젯은 오픈 API를 이용하면 손쉽게 개발될 수 있다. 하지만 위젯의 다양한 응용 가능성에도 불구하고, 현재 오픈 API를 제공하는 웹 사이트가 많지 않다. 그 이유는 웹 사이트 운영자가 오픈 API를 제공하기 위해서는 기존의 웹 서버 구조를 변경하거나 웹 서버 리소스를 수정해야 하는 어려움이 있기 때문이다. 본 논문에서는 기존의 웹 서버 구조 또는 웹 서버 리소스를 변경하지 않고, 개발자가 위젯 개발을 가능하도록 하는 것은 물론, 사용자가 위젯을 사용할 수 있도록 웹 서버를 대신하여 오픈 API를 제공하는 오픈 API 프록시 서버 시스템을 제안한다. 오픈 API 프록시 서버 시스템의 구성은 크게 오픈 API 소스 코드 생성기와 오픈 API 프록시 서버로 이루어져 있다. 오픈 API 소스 코드 생성기는 사용자가 원하는 오픈 API를 생성하도록 사용자에게 GUI를 제공하여 오픈 API 프록시 서버로 오픈 API 소스 코드 생성 요청을 보내는 프로그램이다. 오픈 API 프록시 서버는 본 논문에서 제안하는 HTML 테이블 처리 라이브러리를 이용하여 웹 사이트로부터 HTML 웹 페이지를 받아서 대상 HTML 테이블로부터 유용한 정보를 추출한다. 그리고 이를 XML 문서로 가공하여 오픈 API를 통해 제공한다. 실제 웹 사이트의 HTML 테이블을 대상으로 실험하여 오픈 API 프록시 서버 시스템의 동작을 검증하였다.

키워드 : 위젯, 오픈 API, 프록시 서버 시스템

Abstract A widget is a small application running by the users' favorite services, so they are provided with web contents without explicitly visiting the web site. Although widgets can be easily implemented with Open APIs, only a few web sites provide them because of refactoring the structures of web resource to supply Open APIs to the widget developers. This paper presents an Open API Proxy Server System for widget services. The system consists of two components: an Open API Source Code Generator and an Open API Proxy Server. The Open API Source Code Generator provides a Graphical User Interface (GUI) for users to generate the Open APIs of user's choice and sends the Open API source code generation request to the Open API Proxy Server. The Open API Proxy Server using the HTML Table Processing Library receives the HTML web page from web site and extracts useful information from the target HTML table. The proxy server converts the extracted data into the corresponding XML document which becomes available through the Open API. We verify the operation of the proposed system through experiments with the HTML tables in the example web sites.

Key words : widget, Open API, proxy server system

· 이 논문은 2009년도 광운대학교 교내 학술연구비 지원과 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2010-0000281)

논문접수 : 2010년 2월 4일
심사완료 : 2010년 7월 12일

† 비 회 원 : 광운대학교 임베디드 소프트웨어공학과
iori0316@kw.ac.kr
†† 종신회원 : 광운대학교 컴퓨터공학과 교수
hlee@kw.ac.kr
††† 종신회원 : 광운대학교 정보제어공학과 교수
yhchoi@kw.ac.kr
†††† 비 회 원 : 광운대학교 전자공학과 교수
yuchung@kw.ac.kr

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제9호(2010.9)

1. 서 론

위젯(widget)이란 사용자 기기, 컴퓨터 또는 이동 단말기에 다운로드 하거나 설치할 수 있으며, 로컬 데이터 또는 웹상의 데이터를 갱신 및 보여줌으로써 간편히 쓸 수 있도록 만든 작은 창(window) 형태의 단일 목적 응용프로그램이다[1]. 위젯은 콘텐츠를 제공하는 웹 사이트 등을 직접 방문하지 않고도 사용자가 원하는 콘텐츠를 손쉽게 얻을 수 있어서 이용이 매우 간편하다.

웹 사이트의 콘텐츠 또는 서비스를 제공하는 위젯은 오픈 API(Open Application Programming Interface)를 통해 개발된다. 오픈 API란 웹 2.0 기술 중 대표적인 것으로서, 웹에서 사용할 수 있는 기능을 누구나 사용할 수 있도록 웹 사이트로부터 공개된 API이다. 위젯 개발자는 이러한 오픈 API를 이용해서 위젯을 아주 손쉽게 개발할 수 있으며, 또한 이렇게 만들어진 위젯은 웹 컨텐츠 양을 늘리는데 도움을 주고 새로운 위젯 서비스를 창출해 낼 수 있다. 이렇듯 다양한 위젯 서비스를 위해서 다양한 오픈 API 제공은 필수적이다.

하지만 위젯의 다양한 응용 가능성에도 불구하고, 현재 오픈 API를 제공하는 웹 사이트가 많지 않기 때문에 다양한 위젯 개발 및 사용에 큰 어려움이 따른다. 그 이유는 웹 사이트 운영자가 오픈 API를 제공하기 위해서는 기존의 웹 서버 구조를 변경하거나 웹 서버 리소스를 수정해야 하는 어려움이 있기 때문이다. 따라서 웹 사이트 운영자가 웹 서버 구조나 웹 서버 리소스를 변경하지 않고서도, 손쉽게 이용할 수 있는 오픈 API를 제공하기 위한 기술이 요구된다. 하지만 현재 이러한 문제점을 해결하기 위해 진행된 연구는 전무하다.

따라서 본 논문에서는 기존의 웹 서버 구조 또는 웹 서버 리소스를 변경하지 않고, 프록시 서버(proxy server)[2]를 추가함으로써, 개발자가 위젯 개발을 가능하도록 하는 것은 물론, 사용자가 위젯을 사용할 수 있도록 웹 서버를 대신하여 오픈 API를 제공하는 오픈 API 프록시 서버 시스템을 제안한다. 제안하는 오픈 API 프록시 서버 시스템의 구성은 크게 오픈 API 소스 코드 생성기와 오픈 API 프록시 서버로 이루어져 있다.

오픈 API 소스 코드 생성기는 사용자가 원하는 오픈 API를 생성하도록 오픈 API 프록시 서버로 오픈 API 소스 코드 생성 요청을 보내는 프로그램이다. 이는 사용자가 원하는 웹 페이지의 HTML(HyperText Markup Language)[3] 테이블 영역 선택 및 오픈 API명을 입력할 수 있도록 GUI(Graphical User Interface)를 제공한다. 오픈 API로 제공되는 정보는 HTML 문서상의 테이블이 담고 있는 정보이다. 일반적으로 문서 작성 시 데이터를 정리하여 간단하게 표현하거나, 이를 좀 더 정확하게 전달할 목적으로 테이블을 사용한다. 그러므로 HTML 문서에서 테이블은 중요하고 핵심적인 정보를 포함하고 있다고 볼 수 있으며, 이런 HTML 테이블에서 유용한 정보를 추출하여 오픈 API를 통해 제공하는 것은 가치가 있다.

오픈 API 프록시 서버는 오픈 API 소스 코드 생성기로부터 요청을 받아 해당 오픈 API 소스 코드를 생성한다. 생성된 오픈 API 소스 코드는 차후 위젯으로부터 오픈 API 요청을 받아 이에 대한 응답을 해주는 종단점(end point)이 된다. 이는 내부적으로 본 논문에서 제안하는 *HTML 테이블 처리 라이브러리*를 이용하여 웹 사이트로부터 HTML 웹 페이지를 받아 해당 HTML 테이블을 선택 및 정규화 하여 사용자가 원하는 영역의 데이터를 추출한다. 그리고 이를 XML(eXtensible Markup Language)[4] 문서로 가공하여 오픈 API를 통해 제공한다.

본 논문의 구성은 다음과 같다. 제2장에서는 위젯 및 오픈 API 제공 기술에 대해 서술하고, 제3장에서는 본 연구의 주 내용으로 오픈 API 프록시 서버 시스템의 설계 및 구현에 대해서 서술하고, 제4장에서는 실험을 진행하여 오픈 API 프록시 서버 시스템의 오픈 API 제공 동작을 확인한다. 마지막으로 제5장에서는 본 연구의 결과를 간단히 정리하고 결론 및 향후 과제에 대해서 논의하도록 한다.

2. 위젯 및 오픈 API 제공 기술

2.1 위젯

위젯은 자주 이용하는 서비스를 아이콘 형태로 제작해 독립적으로 구동시키는 작은 응용프로그램이다. 이는 콘텐츠를 제공하는 웹 사이트를 직접 방문하지 않고도 다양한 기능을 이용할 수 있어 사용이 매우 간편하다. 위젯은 특히 개인의 선택에 따라 다양한 조합으로 이용할 수 있기 때문에 개인화 맞춤 서비스의 기반이 되고 있다. 이러한 위젯은 크게 데스크탑 위젯, 웹 위젯, 모바일 위젯 세 종류로 분류할 수 있으며 각각의 특징은 다음과 같다.

*데스크탑 위젯*은 호스트 소프트웨어 시스템(위젯 엔진) 내에서 작동하는 작은 응용프로그램이다[5]. 가장 기본적인 데스크탑 위젯으로는 계산기, 시계, 액자, 또는 메모장 위젯 등이 있으며, 이는 컴퓨터 바탕화면에 띄어놓고 바로 이용할 수 있으므로 자주 이용하는 서비스로의 접근 경로를 단축시키는 효과가 있다. 데스크탑 위젯은 웹 위젯에 비해 시각적 요소가 뛰어나고 속도가 빠르지만, 설치 기반이라 사용이 불편하고 플랫폼에 종속적인 한계를 가진다.

웹 위젯은 HTML 기반의 웹 페이지 내에서 단일 기

능을 수행하는, 삽입과 삭제가 가능한 한 단위의 코드 묶음이다[5]. 웹 위젯은 초기 광고, 마케팅, 이미지, 동영상 등 컨텐츠 배포 매체 형태로 등장하여 진화하였으며, 또한 웹 서비스로의 연결 통로 및 컨텐츠 소비 채널 역할을 하였다. 웹 위젯은 데스크탑 위젯에 비해 시각적 요소가 떨어지고 속도가 느리지만, 인터넷 웹 브라우저 상에서 동작하므로 별도의 설치가 필요 없고 데이터베이스가 웹에 있어 동기화가 필요 없다는 강점이 있다.

모바일 위젯은 모바일 단말기 상의 위젯 엔진 위에서 구동되는 작은 크기의 응용프로그램으로서, 모바일 단말기 대기화면에서 상시적으로 동작하는 위젯이다[5]. 모바일 위젯은 모바일 단말기 플랫폼에 종속적인 한계를 갖지만, LBS(Location-Based Service), MMS(Multimedia Messaging Service) 연동 등 무선 환경에서만 접근 가능한 인프라 및 서비스를 매시업(mash-up)할 수 있어 기존의 웹/데스크탑 위젯이 보여주지 못하던 기능을 수행할 수 있다. 특히 위젯 이용에 대한 과금 체계가 매우 간편하여 유료화에 있어서도 큰 가능성을 가지고 있다.

2.2 오픈 API 제공 기술

오픈 API는 전통적인 API의 개념을 웹으로 확장한 것으로서, 인터넷 서비스 회사가 자사에서 독점적으로 사용하던 서비스, 정보 등을 외부에 개방하여 사용자로 하여금 웹 사이트에 접속해서 필요한 데이터를 요청하고 응답 받을 수 있게 해주는 API이다. 이러한 오픈 API를 제공하는 기술들은 REST(Representational State Transfer)[6], XML-RPC(eXtensible Markup Language Remote Procedure Call)[7], SOAP(Simple Object Access Protocol)[8] 등이 있으며 각각의 특징은 다음과 같다.

REST는 분산 컴퓨팅 플랫폼 모델이며, 세계에서 가장 큰 분산 응용인 웹에서 사용하고 있는 웹 구조 스타일 모델이다. 현재의 웹의 기본 요소는 URI, HTTP, HTML, XML이며, REST는 이러한 인터넷 표준만을 이용한다. REST에서 리소스의 식별은 URI로, 상태는 이를 표현한 문서이며 이는 HTTP를 통해 전달된다. 리소스의 내용은 XML로 기술하며, 리소스의 참조, 생성, 수정, 삭제에는 HTTP의 표준 메서드인 GET, PUT, POST, DELETE 등을 이용하여 분산 컴퓨팅을 모델링(modeling)한다.

XML-RPC는 분산 컴퓨터 환경에서 이기종의 컴퓨터 자원을 사용하는 기술인 RPC(Remote Procedure Call)의 개념을 웹으로 옮겨온 것으로서, 프로그램에서 함수 호출을 하듯이 원격에 있는 사이트에 정보를 요청하고 받아올 수 있다. 이때 주고받는 인자와 반환 값은 XML 데이터로 인코딩하고, 이 데이터를 HTTP(POST 메서드 사용)를 통해 전달한다. XML-RPC로 인코딩할 수 있는 정보는 배열, Base64로 인코딩된 바이너리, 이진 값, 날짜, 시간, 실수, 정수, 문자열, 구조체로 한정되어 있다.

SOAP는 XML-RPC가 발전한 프로토콜이다. 앞서 소개한 XML-RPC는 기본적인 데이터 타입만을 표현할 수 있기 때문에 복잡한 사용자 정의 타입을 정의하는 데는 어려움이 따른다. 이를 극복하고자 좀 더 복잡한 데이터 구조를 표현하고 이에 특화된 처리를 지원하기 위한 프로토콜이 SOAP이다. SOAP는 REST나 XML-RPC에 비해 복잡도가 상당히 높으며, 주로 엔터프라이즈 솔루션의 웹 서비스 구현에 자주 사용되고 있다.

SOAP는 모든 객체를 관리하는 한 개의 종단점만을 참조하는 하나의 URI만을 기술한다. 따라서 SOAP의 종단점과 매핑되는 내부 리소스는 직접 주소지정이 불가능하고, 이들 리소스 조작을 위해 독자적인 메서드들을 제공한다. 그러나 이러한 메서드들은 HTTP의 CRUD(Create, Read, Update and Delete) 같은 표준 메서드(PUT, GET, POST, DELETE 등)가 아니기 때문에 클라이언트는 반드시 이러한 메서드들을 이해하고 있어야 한다. 만약 추후 기능 확장 등을 위해서 메서드들 내용이 변경되어야 하는 경우, 클라이언트는 이 변경된 메서드를 이용하여야 하기 때문에 기존 클라이언트 응용 프로그램의 변경이 불가피하고, 이러한 점에서 확장성이 부족하다. 따라서 SOAP 클라이언트는 웹 서비스의 인터페이스를 알아야 하며, 이것은 변경될 가능성이 있으므로 반드시 현재의 인터페이스를 알아야 한다. 물론 응용마다 제공되는 이러한 메서드 인터페이스에 대해서는 WSDL(Web Services Description Language)[9]에 기술되며, 이는 개별적으로 또는 UDDI(Universal Description Discovery and Integration)[10]를 통해 인터페이스에 대한 정보 획득이 가능하다. 그러나 UDDI는 확장성이 부족하다.

본 논문에서는 오픈 API를 제공하기 위해 확장성이 뛰어나고 구현하기가 용이한 REST-RPC 하이브리드 아키텍처[11]를 기반으로 프록시 서버를 설계하였다.

3. 오픈 API 프록시 서버 시스템의 설계

본 장에서는 기존에 오픈 API를 통해 제공되지 않은 웹 컨텐츠를 오픈 API를 통해 제공될 수 있도록 하는, 위젯 서비스를 위한 오픈 API 프록시 서버 시스템을 제안한다. 이는 기존의 웹 서버 구조 또는 웹 서버 리소스를 변경하지 않고 기존 웹 서버에 위젯 서비스를 위한 프록시 서버 시스템을 그림 1과 같이 추가하여, 이 프록시 서버 시스템으로 하여금 오픈 API를 제공함으로써, 개발자가 위젯 개발을 가능하도록 하는 것은 물론, 사용자가 위젯을 사용할 수 있도록 하는 시스템이다.

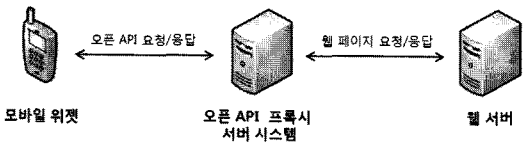


그림 1 프록시 서버 시스템을 통한 오픈 API 요청과 응답

위젯 서비스를 위한 오픈 API 프록시 서버 시스템의 구성은 크게 오픈 API 소스 코드 생성기와 오픈 API 프록시 서버로 이루어져 있다. 오픈 API 소스 코드 생성기를 통해 생성된 오픈 API는, 프록시 서버 시스템의 HTML 테이블 처리 라이브러리를 이용하여 대상 HTML 테이블 상의 유용한 정보 추출 및 이를 XML 문서로 가공하여 제공하는 역할을 수행한다. 이들에 대한 자세한 설명은 다음과 같다.

3.1 오픈 API 소스 코드 생성기 설계

오픈 API 소스 코드 생성기는 사용자에게 GUI를 제공함으로써 사용자가 원하는 오픈 API를 손쉽게 만들 수 있도록 하기 위한 프로그램이다. 그림 2는 오픈 API 소스 코드 생성기를 이용한 오픈 API 생성 과정을 보여준다.

오픈 API 소스 코드 생성기는 사용자에게 웹 브라우저와 같은 HTML 뷰어를 제공하여, 사용자가 원하는 정보 즉, 오픈 API를 통해 제공할 정보가 있는 웹 페이지로 이동하는 것을 돕는다. 사용자가 웹 페이지 상의 대상 HTML 테이블을 마우스 클릭하여 선택하면, HTML 테이블이 정규화되어 화면에 출력된다.

HTML 테이블 정규화는 각 셀의 정확한 행과 열의 위치를 파악하기 위해 필요하다. 만약 *td* 또는 *td* 요소(element)의 *colspan*(또는 *rowspan*) 속성(attribute) 값이 2 이상의 k 값을 갖는다면, 그 요소는 현재 행 방향

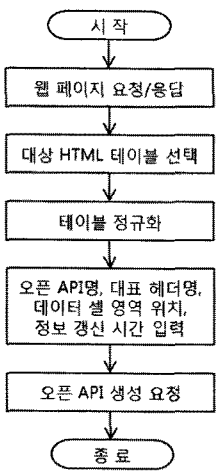


그림 2 오픈 API 생성 과정

지점	기온(℃)		바람	
	현재 기온	체감 온도	풍향	풍속(m/s)
서울	-13.3	-18.5	북북서	2.3
인천	-11.3	-16.8	북동	2.8

테이블 정규화

지점	기온(℃)	기온(℃)	바람	바람
지점	현재 기온	체감 온도	풍향	풍속(m/s)
서울	-13.3	-18.5	북북서	2.3
인천	-11.3	-16.8	북동	2.8

그림 3 HTML 테이블 정규화 예

(또는 열 방향)으로 $k-1$ 만큼 더 확장되어 있는 것을 의미한다. 이러한 각 요소들의 *colspan*과 *rowspan* 속성 값에 따라 여분의 셀을 삽입함으로써 HTML 테이블을 정규화 할 수 있다. 그림 3은 HTML 테이블 정규화의 예를 보여 준다.

그 후, 정규화된 HTML 테이블의 각 셀마다 행과 열의 위치 정보를 나타내기 위한 *posRow*와 *posCol* 이라는 속성을 정의하여 해당하는 행과 열의 위치 값을 각각 할당한다. 그림 3의 정규화된 하단 HTML 테이블을 예로 들면, '서울' 셀은 세 번째 행, 첫 번째 열에 위치하므로 *posRow* 값은 3, *posCol* 값은 1이 된다.

사용자는 오픈 API 소스 코드 생성기에서 제공하는 GUI를 통해, 오픈 API명, 데이터 셀들을 대표하는 헤더명, 추출할 데이터 셀 영역 위치, 그리고 정보 갱신 시간을 입력하여 오픈 API 프록시 서버로 오픈 API 생성 요청을 보낸다.

3.2 오픈 API 프록시 서버 설계

오픈 API 프록시 서버는 웹 사이트로부터 HTML 웹 페이지를 받아서 대상 HTML 테이블로부터 유용한 정보를 추출하고 이를 XML로 가공하여 오픈 API를 통해 제공하는 프록시 서버이다.

오픈 API 프록시 서버의 구조는 그림 4와 같다. 웹 서버는 HTTP 프로토콜을 지원하여 서버 역할을 수행한다. 이는 위젯으로부터 오픈 API를 요청 받고 요청 URI(Uniform Resource Identifier)를 탐색하여 요청 문서가 어떤 언어로 작성된 문서인지 판단한다. 요청 문서가 클라이언트 측 스크립트 문서(HTML, CSS, JavaScript 등)라면 웹 서버가 이를 직접 처리하여 응답한다. 또한, 요청 문서가 서버 측 스크립트 문서(PHP, ASP, JSP 등)라면 이를 서버 측 스크립트 해석 엔진에게 해석을 의뢰한다. 서버 측 스크립트 해석 엔진은 서버 측 스크립트로 작성된 오픈 API를 해석하여 실행하는 역할을 수행한다. HTML 테이블 처리 라이브러리에

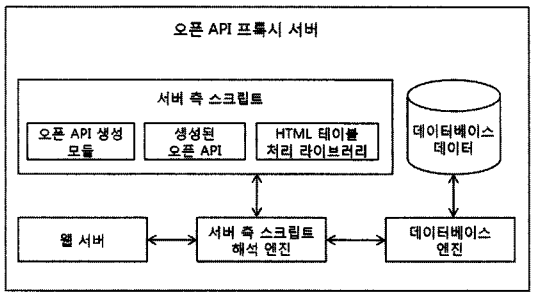


그림 4 오픈 API 프록시 서버 구조

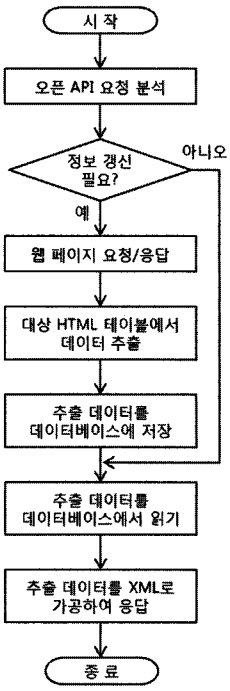


그림 5 오픈 API 제공 과정

의해 대상 HTML 테이블로부터 추출된 데이터는 데이터베이스에 저장되며, 이는 데이터베이스 엔진에 의해 조작되고 관리된다.

프록시 서버의 오픈 API 생성 모듈은 오픈 API 소스 코드 생성기로부터 오픈 API 생성 요청을 받아 오픈 API 소스 코드를 생성한다. 생성된 오픈 API 소스 코드 즉, 오픈 API 종단점은 위젯으로부터 오픈 API 요청을 받았을 때, 내부적으로 HTML 테이블 처리 라이브러리의 기능을 이용하여 그림 5와 같이 오픈 API를 제공한다.

오픈 API 종단점은 위젯으로부터 받은 오픈 API 요청을 분석하여 정보 갱신이 필요한지 아닌지를 판단한다. 이전 추출 데이터가 데이터베이스에 캐싱(caching)된 시간이 아직 다음 정보 갱신 시간을 초과하지 않았다면 이를 바로 XML로 가공하여 응답하고, 정보 갱신 시간을 초과하였다면 최신 추출 데이터를 얻기 위해 정보를 갱신한다. 오픈 API 종단점은 정보를 갱신하기 위해 먼저 해당 웹 페이지를 요청하여 응답받는다. 그 후, 수신된 웹 페이지 상의 대상 HTML 테이블에서 데이터를 추출하고 이를 데이터베이스에 저장한다. 이는 추출 데이터를 데이터베이스에 캐싱함으로써 오픈 API 요청에 대한 응답 처리 속도를 높이기 위함이다. 오픈 API 종단점은 오픈 API 요청에 응답하기 위해 데이터베이스로부터 추출 데이터를 읽어 XML 문서로 가공하여 이를 위젯에게 제공한다.

3.3 HTML 테이블 처리 라이브러리 설계

웹 문서 표준인 HTML은 웹 문서를 시각적으로 렌더링하기 위한 포맷이기 때문에 컴퓨터로 하여금 정보를 처리하게 한다는 측면에서 한계를 갖는다. 반면 XML은 논리적 구조 정보를 표현할 수 있으며 플랫폼에 독립적이라는 장점 때문에 다양한 분야에서 정보의 공유 및 교환을 위한 표준으로 널리 사용되고 있다. 따라서 HTML 문서로부터 유용한 정보를 추출하여 이를 XML 문서로 변환하는 방법이 필요하다.

본 논문에서는 HTML 테이블 처리 라이브러리를 제안한다. 이는 대상 HTML 테이블로부터 데이터를 추출하여 이를 XML 문서로 변환하는 역할을 수행한다. 또한, 이는 2차원 형태의 HTML 테이블 만을 대상으로 한다.

대상 HTML 테이블로부터 데이터를 추출하는 과정은 다음 Algorithm 1과 같다.

Algorithm 1 Extracting data from the HTML table.

Input: the normalized table T,
the set of the header names H,
the set of the starting indexes for each row RS,
the set of the ending indexes for each row RE,
the set of the indexes for each column C.
Output: the set of the extracted data D.

Consider the number of elements k of H
FOR the index i = 1 to k
Consider current header name H_i in H,
the current starting index RS_i in RS,
the current ending index RE_i in RE,
the current index C_i in C
FOR the index of row r = RS_i to RE_i DO
D[H_i][i] = getCellValue(T[r][C_i])
ENDFOR
ENDFOR

```
getCellValue(element E) /* extract the value
                           from the td element */
{
  WHILE E has its child nodes DO
    Consider the first child node C of E,
      the value V of C
    IF C exists and V exists THEN
      RETURN with V
    ENDIF
    E = C
  ENDWHILE
}
```

정규화된 대상 HTML 테이블 T는 구조적으로 이차원 배열과도 같다. 이러한 특징을 이용하여, 해당 셀 영역 내에 위치하는 데이터를 추출한다. 대표 헤더명과 추출한 데이터를 키-값 쌍(key-value pair)으로 하여 연관 배열 형식의 자료구조 D에 저장한다.

자료구조 D에 저장된 추출 데이터를 XML 문서로 변환하는 과정은 다음 Algorithm 2와 같다.

Algorithm 2 Converting the extracted data into its corresponding XML document.

Input: the Open API name N,
the set of the extracted data D.
Output: the XML document X.

Consider the number of values k that are included
in a header name field of D,
the number of header name fields l of D
X = <?xml version="1.0" encoding="euc-kr" ?>
X += <N>
FOR the index i = 0 to k DO
FOR the index j = 0 to l DO
Consider the current header name of the
field H_j
X += <H_j>D[H_j][i]</H_j>
ENDFOR
ENDFOR
X += </N>

오픈 API명 N을 XML문서 X의 최상위 요소(root element)명으로 설정하고, 자료구조 D에 저장된 대표 헤더명 H_j와 이에 대응하는 추출 데이터 값 D[H_j][i]를 각각 XML 문서의 요소명과 그 요소의 값으로 설정하여 XML 문서로 가공한다.

지점	현재 기온	체감 온도	풍향
서울	10.2	9.5	동북
인천	11.6	9.6	동북동
부산	12.9	11.9	동남동

그림 6 대상 HTML 테이블 예

표 1 입력 파라미터 예

오픈 API명	get_weather		
대표 헤더명 (H)	추출할 데이터의 셀 영역 위치		
	행 시작 첨자 (RS)	행 끝 첨자 (RE)	열 첨자 (C)
H[1] = 지점	RS[1] = 2	RE[1] = 3	C[1] = 1
H[2] = 기온	RS[2] = 2	RE[2] = 3	C[2] = 2

그림 6의 대상 HTML 테이블을 예로 들어 Algorithm 1과 Algorithm 2의 동작 과정을 설명하겠다. 그림 6의 대상 HTML 테이블에 사각 점선으로 표시된 부분은 HTML 테이블로부터 데이터를 추출하기 위한 셀 영역을 나타낸다.

표 1은 오픈 API 소스 코드 생성 요청을 통해 전달받은 오픈 API 명, 대표 헤더명, 그리고 추출할 데이터의 셀 영역 위치(행 시작 위치, 행 끝 위치, 열 위치)에 대한 입력 파라미터의 예를 나타낸다.

Algorithm 1은 그림 6의 HTML 테이블 T로부터 데이터를 추출하여 자료구조 D에 저장한다. 대표 헤더명 '지점'(H[1])에 대한 데이터가 있는 셀 영역 위치가 두 번째 행(RS[1])부터 세 번째 행(RE[1])까지, 그리고 첫 번째 열(C[1])이라면, D['지점'][1]에 T[2][1]의 값인 '서울'을 저장하고 D['지점'][2]에 T[3][1]의 값인 '인천'을 저장한다. 마찬가지로 대표 헤더명 '기온'(H[2])에 대한 데이터가 있는 셀 영역 위치가 두 번째 행(RS[2])부터 세 번째 행(RE[3])까지, 그리고 두 번째 열(C[2])이라면, D['기온'][1]에 T[2][2]의 값인 '10.2'를 저장하고 D['기온'][2]에 T[3][2]의 값인 '11.6'을 저장한다.

Algorithm 2는 자료구조 D에 저장된 추출 데이터를 XML 문서로 변환한다. 먼저 오픈 API 명 'get_weather'를 XML문서의 최상의 요소로 설정한다. 이어서 자료구조 D에 저장된 각각의 대표 헤더명 '지점'과 '기온'을 XML 문서의 요소명으로 설정하고, 이에 대응하는 각각의 추출 데이터 값 '서울'과 '10.2', 그리고 '인천'과 '11.6'을 그 요소의 값으로 설정한다. 이렇게 가공된 XML 문서는 다음과 같다.

```
<get_weather>
    <지점>서울</지점>
    <기온>10.2</기온>
    <지점>인천</지점>
```

<기온>11.6</기온>
</get_weather>

4. 실 험

4.1 오픈 API 프록시 서버 시스템 구현

위젯 서비스를 위한 오픈 API 프록시 서버 시스템은 오픈 API 소스 코드 생성기와 오픈 API 프록시 서버로 나뉘어 구현되었다.

오픈 API 소스 코드 생성기는 윈도우 기반의 Microsoft Visual C++과 JavaScript 언어를 사용하여 구현되었다. 이는 오픈 API를 생성하고자 하는 사용자가 웹 브라우저 화면을 통해 원하는 웹 페이지로 이동할 수 있도록 하기 위해 Microsoft에서 제공하는 Internet Explorer HTML 뷰어 컴포넌트를 이용하였고, 사용자가 원하는 HTML 테이블을 선택할 수 있도록 Microsoft에서 제공하는 DOM(Document Object Model)[12] 라이브러리를 이용하여 구현되었다. 또한, HTML 테이블 정규화 기능, 사용자로부터 오픈 API 생성에 필요한 정보들을 입력받는 GUI, 오픈 API 생성 요청 기능은 JavaScript로 구현되었다.

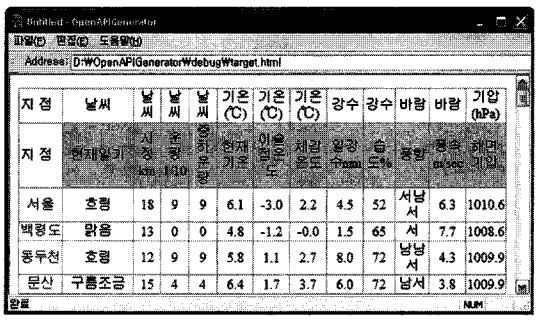
오픈 API 프록시 서버를 구현하기 위해, HTTP 프로토콜을 지원하는 대표적인 웹 서버인 Apache[13] 웹 서버, 서버 측 스크립트 언어인 PHP[14]를 해석하기 위한 PHP 엔진, 그리고 데이터베이스 데이터를 저장하고 관리하기 위한 MySQL[15] 데이터베이스 관리 시스템을 호스트 컴퓨터에 설치하였다. 또한, 오픈 API 소스 코드 생성 모듈과 HTML 테이블 처리 라이브러리는 PHP를 사용하여 구현되었으며, 프록시 서버와 웹 사이트간의 HTTP 통신 기능, MySQL 데이터베이스 조작 기능은 PHP 확장 라이브러리를 이용하여 구현되었다.

4.2 실험 수행 및 결과

본 절에서는 구현한 오픈 API 프록시 서버 시스템의 동작을 검증하기 위해 실제 웹 사이트의 HTML 테이블을 대상으로 실험을 진행하였다. 실험 대상 사이트는 유용한 정보를 제공하지만 이에 대한 오픈 API를 별도로 제공하지 않는 기상청 웹 사이트, 부동산 웹 사이트, 그리고 고속버스 웹 사이트를 선택하였다.

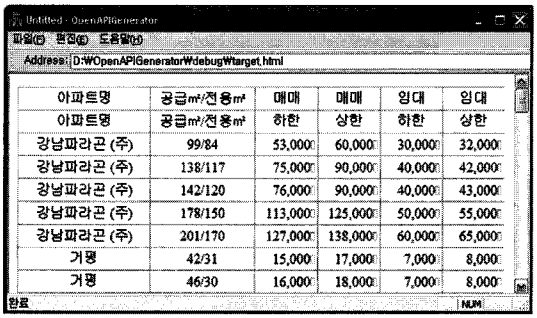
기상청 웹 사이트의 지점별 날씨 정보를 제공하는 'get_weather' 오픈 API, 부동산 웹 사이트의 아파트 시세 정보를 제공하는 'get_apartment_price' 오픈 API, 그리고 고속버스 웹 사이트의 노선별 요금 정보를 제공하는 'get_bus_charge' 오픈 API를 각각 생성하여, 이를 요청 및 응답 받아 동작을 확인해 보기로 하였다.

그림 7의 (a), (b), 그리고 (c)는 오픈 API 소스 코드 생성기를 통해 선택한 각 웹 사이트의 정규화된 대상 HTML 테이블을 나타낸다.



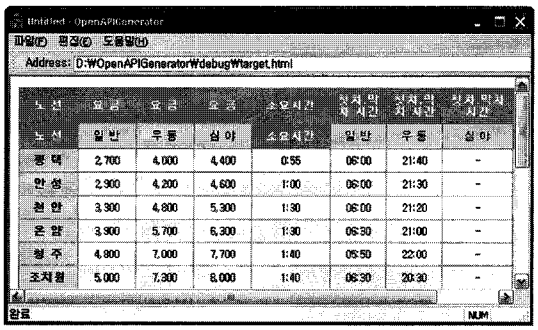
지 점	날씨	날씨	날씨	기온(℃)	기온(℃)	기온(℃)	강수	강수	바람	바람	기압(hPa)
지 점	현재일기	시정 km	운량 1/10	중하운량	현재기온	이슬점온도	체감온도	왕강수 mm	습도 %	풍향	풍속 m/sec	해면기압
서울	흐림	18	9	9	6.1	-3.0	2.2	4.5	52	서남서	6.3	1010.6
백령도	맑음	13	0	0	4.8	-1.2	-0.0	1.5	65	서	7.7	1008.6
동두천	흐림	12	9	9	5.8	1.1	2.7	8.0	72	남남서	4.3	1009.9
문산	구름조금	15	4	4	6.4	1.7	3.7	6.0	72	남서	3.8	1009.9

(a)



아파트명	공급㎡/전용㎡	매매	매매	임대	임대
아파트명	공급㎡/전용㎡	하한	상한	하한	상한
강남파라곤 (주)	99/84	53,000	60,000	30,000	32,000
강남파라곤 (주)	138/117	75,000	90,000	40,000	42,000
강남파라곤 (주)	142/120	76,000	90,000	40,000	43,000
강남파라곤 (주)	178/150	113,000	125,000	50,000	55,000
강남파라곤 (주)	201/170	127,000	138,000	60,000	65,000
거평	42/31	15,000	17,000	7,000	8,000
거평	46/30	16,000	18,000	7,000	8,000

(b)

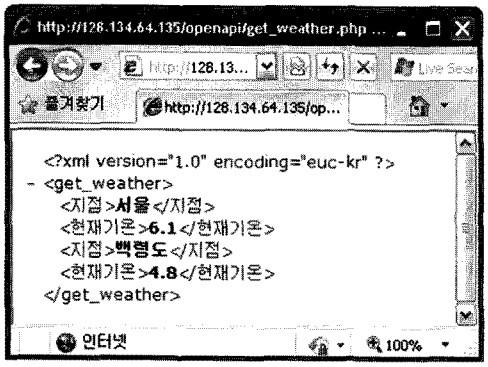


노선	요금	요금	요금	소요시간	첫차 출발 시간	첫차 출발 시간	첫차 출발 시간
노선	일반	우등	심야	소요시간	일반	우등	심야
광 덕	2,700	4,000	4,400	0:55	06:00	21:40	-
안 성	2,900	4,200	4,600	1:00	06:00	21:30	-
천 안	3,300	4,800	5,300	1:30	06:00	21:20	-
온 양	3,900	5,700	6,300	1:30	06:30	21:00	-
청 주	4,800	7,000	7,700	1:40	05:50	22:00	-
조치원	5,000	7,300	8,000	1:40	06:30	20:30	-

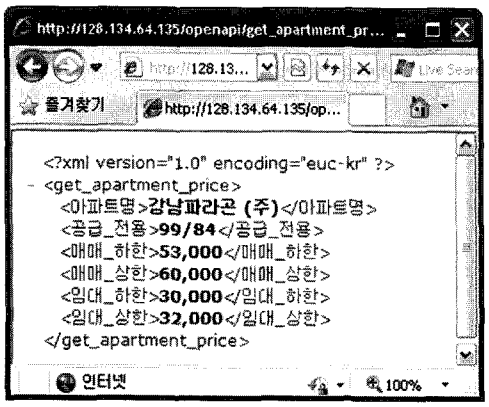
(c)

그림 7 (a) 기상청 웹 사이트의 대상 HTML 테이블
(b) 부동산 웹 사이트의 대상 HTML 테이블
(c) 고속버스 웹 사이트의 대상 HTML 테이블

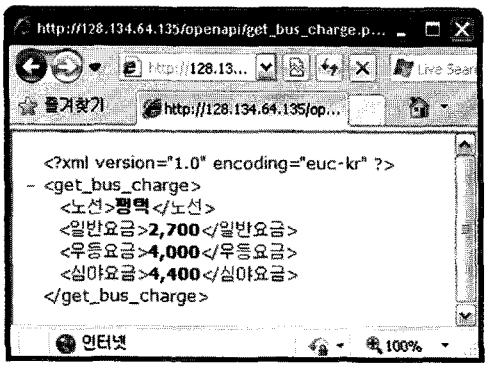
오픈 API 소스 코드 생성기의 GUI를 통해, 오픈 API 명, 정규화 된 대상 HTML 테이블 상에서 추출할 각 데이터 셀 영역 위치, 대표 헤더명, 그리고 정보 갱신 시간을 입력하여 오픈 API 프록시 서버로 오픈 API 소스 코드 생성 요청을 보낸다. 오픈 API 프록시 서버의 오픈 API 소스 코드 생성 모듈은 이 요청을 받아, 전달받은 오픈 API 생성 정보를 이용하여 오픈 API 소스 코드인 PHP 스크립트 파일을 생성한다. 이 생성된 PHP 스크립트 파일은 차후 위젯으로부터 오픈 API 요청을 받아 이에 대한 응답을 해주는 종단점이 된다.



(a)



(b)



(c)

그림 8 (a) get_weather API 요청에 대한 XML 응답 (b) get_apartment_price API 요청에 대한 XML 응답 (c) get_bus_charge API 요청에 대한 XML 응답

그림 8의 (a), (b), 그리고 (c)는 각 생성된 오픈 API를 요청하여 응답받은 XML 문서의 예를 보여준다.

다음은 'get_weather' 오픈 API를 이용하는 날씨 위젯을 구현하여, 이를 Opera 웹 브라우저[16]의 위젯 엔진을 이용하여 실행하였다. 날씨 위젯은 오픈 API 프록시 서버를 통해 'get_weather' 오픈 API를 요청하고 응답받는다. 그림 9는 날씨 위젯의 실행 화면을 보여준다.

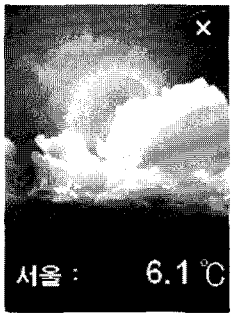


그림 9 오픈 API를 이용하는 날씨 위젯

5. 결 론

본 논문은 웹 서버를 대신하여 오픈 API가 제공되지 않은 웹 컨텐츠에 대한 오픈 API를 제공하는 오픈 API 프록시 서버 시스템을 제안하였다. 오픈 API 프록시 서버 시스템의 동작을 검증하기 위해 실제 웹 사이트의 HTML 테이블을 대상으로 실험을 수행하였고, 실험해 본 결과, 오픈 API 프록시 서버 시스템을 통해 위젯 서비스가 가능하다는 것을 확인하였다. 위젯 개발자는 오픈 API 프록시 서버 시스템을 통해 제공되는 오픈 API를 이용하여 다양하고 창의적인 위젯을 개발할 수 있으며, 이를 통해 웹 콘텐츠 양이 많아지게 되고 새로운 위젯 서비스가 많이 창출될 것으로 기대된다.

본 논문에서 제안한 오픈 API 소스 코드 생성기에서, 사용자는 대표 헤더명과 추출할 데이터 셀 영역 위치를 수동으로 입력하였다. 향후 본 연구에서는 이러한 과정을 자동으로 처리하기 위해, HTML 테이블의 데이터 셀과 이에 대응하는 헤더 셀을 자동으로 식별하여 처리하는 방법에 관한 연구를 진행할 계획이다.

참 고 문 헌

[1] W3C, "Widgets 1.0: The Widget Landscape (Q1 2008)," http://www.w3.org/TR/widgets-land

[2] K. W. Ross, and J. F. Kurose, *Computer Networking: A Top-Down Approach*, 4th Ed., pp.134-138, Addison-Wesley Professional, 2007.

[3] W3C, "HTML 4.01 Specification," http://www.w3.org/TR/html401

[4] W3C, "Extensible Markup Language(XML) 1.0," http://www.w3.org/TR/xml

[5] C. Kaar, "An Introduction to Widgets with Particular Emphasison Mobile Widgets," http://www.symbianresources.com/tutorials/techreports/widgets/kaar07widgets.pdf

[6] R.T. Fielding, *Architectural Styles and the Design of Network-Based Software Architectures*, PhD dissertation, Dept. of Computer Science, Univ. of California, Irvine, Calif., 2000.
[7] XML-RPC homepage, http://www.xmlrpc.com
[8] W3C, "SOAP Version 1.2 Part 0: Primer (Second Edition)," http://www.w3.org/TR/2007/REC-soap12-part0-20070427
[9] W3C, "Web Services Description Language (WSDL) Version 2.0 Part 0: Primer," http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626
[10] UDDI homepage, http://uddi.org
[11] L. Richardson, and S. Ruby, *Restful Web Services*, pp.46-48, O'REILLY, 2007.
[12] W3C, "Document Object Model (DOM) Level 1 Specification," http://www.w3.org/TR/REC-DOM-Level-1
[13] Apache web server homepage, http://www.apache.org
[14] PHP server-side scripting language homepage, http://www.php.net
[15] MySQL database homepage, http://www.mysql.com
[16] Opera web browser homepage, http://www.opera.com

안 병 현
2008년 안양대학교 컴퓨터공학과 학사. 2008년~현재 광운대학교 임베디드 소프트웨어공학과 석사과정. 관심분야는 웹 서비스, 웹 테이블 마이닝

이 혁 준
1987년 University of Michigan, Computer Science 학사. 1989년 Syracuse University, Computer Science 석사. 1993년 Syracuse University, Computer Science 박사. 1994년~1996년 삼성전자(주) 멀티미디어 연구소 선임연구원. 1996년~현재 광운대학교 컴퓨터공학과 교수. 관심분야는 모바일 컴퓨팅, 무선 네트워크, 차량간 통신

최 용 훈
1995년 연세대학교 전자공학과 학사. 1997년 연세대학교 전자공학과 석사. 2001년 연세대학교 전기전자공학과 박사. 2001년~2002년 University of Maryland, Research Associate. 2002년~2005년 LG 전자 책임연구원. 2005년~현재 광운대학교 정보제어공학과 부교수. 관심분야는 통신 네트워크

정 영 욱
1997년 한국과학기술원 전기 및 전자공학 학사. 1999년 한국과학기술원 전기 및 전자공학 석사. 2003년 한국과학기술원 전자전산학 박사. 2005년~현재 광운대학교 전자공학과 조교수. 관심분야는 차세대 이동통신/무선인터넷 시스템 무선자원관리, 위치 기반 이동성 관리, 3D over IP network, ITS