

## 음성인식을 위한 웹페이지 변환 웹서비스와 음성라이브러리 구현

An Implementation of the Speech-Library and Conversion  
Web-Services of the Web-Page for Speech-Recognition

오지영, 김윤중  
한밭대학교

Oh Jee-Young, Kim Yoon-Joong  
Hanbat National University

### 요약

본 연구에서는 음성인식을 위한 웹페이지 변환 웹서비스와 음성을 녹음하고 전송하는 음성라이브러리를 구현하였다. 구현된 시스템은 웹서비스 소비자 및 웹서비스 제공자들로 구성되어 있다. 웹서비스 소비자는 음성을 녹음하고 웹서비스를 호출하여 음성인식을 요청한 후 결과를 사용자에게 반환하는 기능을 한다. 웹서비스 소비자는 음성라이브러리(Speech-Library)와 웹서비스와 통신하는 프록시라이브러리를 포함한다. 음성라이브러리는 사용자가 녹음한 음성에서 음성데이터만 추출하는 전처리 과정과 사용자의 음성과 매핑되는 링크를 검색하는 기능을 수행한다. 프록시라이브러리의 기능은 두개의 웹서비스를 호출하고 반환되는 결과값을 수신받는다. 웹서비스 제공자는 파싱 웹서비스와 음성인식 웹서비스로 구성되어 있다. 파싱 웹서비스는 일반 웹페이지를 ActiveX 컨트롤을 삽입하여 음성인식이 가능한 웹페이지로 재구성한다. 음성인식 웹서비스는 기존의 연구에서 구현된 시스템을 사용하였다. 실험 결과, 일반 웹페이지를 재구성하고 링크 테이블을 생성한 것을 확인할 수 있었다. 또한 사용자의 음성과 매핑되는 URL을 검색하는 것도 확인하였다. 또한 음성인식 웹서비스의 결과에 매핑되는 URL을 검색하여 사용자에게 웹페이지를 반환하는 것도 확인하였다.

### Abstract

This paper implemented speech-library and the Web Services that conversion the Web page for the speech recognition. The system is consisted of Web services consumer and Web services providers. The Web services consumer has libraries that Speech-library and proxy-library. The Speech-library has functions as follows: from the user's speech extracted speech-data and searching the URL in link-table that is mapped with user's speech. The proxy-library calls two web services and is received the returning result.

The Web services provider consisted of Parsing Web Services and Speech-Recognition Web Services. Parsing Web Services adds ActiveX control and reconstructs web page using the speech recognition. The speech recognizer is the web service providers that implemented in the previous study. As the result of experiment, we show that reconstructs web page and creates link-Table. Also searching the URL in link-table that is mapped with user's speech. Also confirmed returning the web page to user by searching URL in link-table that is mapped with the result of speech recognition web services.

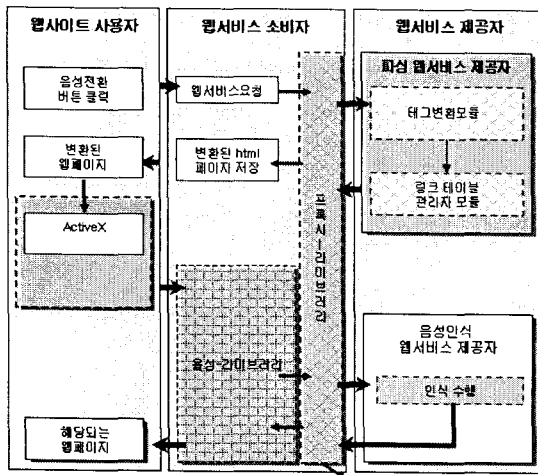
## I. 서론

기존에 인터넷에서 제공하는 정보는 대부분 텍스트였지만 오늘날은 많은 멀티미디어 정보를 포함한다. 이와 같은 텍스트와 멀티미디어 정보들은 마우스나 키보드를 클릭함으로써 정보를 이용할 수 있다. 이렇듯 인터넷과 관련된 제반 기술이 급격히 발전함에 따라 음성인식 기술을 적용하고자 하는 욕구가 증대되고, 일부 음성인식기술이 이용되는 단계에 이르고 있다.

본 연구에서는 인간과 웹페이지의 통신 수단으로 음성을 이용하기 위한 방법을 제안한다. 이미 웹에 관련된 음성인식시스템에 대한 연구가 활발히 진행되고 있다. 1988년에 '웹 브라우저 상에서 한국어 음성인식을 이용한 정보 검색 시스템'[1]이

개발되었으며 2001년에는 '한국어 음성 웹브라우저 설계 및 구현'[2]이 발표된 바 있다. 이들 연구의 음성인식 시스템이 웹에 적용되기 위해서 각각의 웹사이트에 음성 데이터베이스를 구축하거나 음성인식용 웹사이트를 재개발해야 했다. 또는 별도의 음성인식용 웹브라우저를 설치해야 하는 등 많은 비용과 시간을 투자해야 하는 불편함을 지니고 있다.

본 연구에서는 기존에 구현한 '음성인식용 웹페이지 변환을 위한 웹서비스 구현'[6]을 기초로 하여 웹서비스 제공자를 사용하기 위하여 필요한 프록시와 음성데이터 전처리 과정을 라이브러리로 구현하여 웹 개발자가 웹서비스와 음성인식기술을 숙지하지 않아도 쉽게 사용할 수 있는 방안을 제안한다.



▶▶ 그림 4. 구현된 시스템

본 연구에서 구현한 시스템은 그림 1과 같이 구성되어 있다. 첫 번째 웹사이트를 이용하고 음성을 녹음하는 웹사이트 사용자와 두 번째 웹페이지를 제공하고 음성인식기능을 필요로 하는 웹서비스 소비자, 마지막으로 웹서비스 제공자로 구성되어 있다. 웹서비스 제공자는 웹페이지를 분석하고 변환시키는 파싱 웹서비스 제공자와 인식기능을 제공하는 음성인식 웹서비스 제공자로 구성되어 있다. 이와 같은 구성에서 음성인식 웹서비스 제공자는 기존의 연구에서 구현한 시스템을 이용하였다.

본 논문의 구성은 1장에서 서론을 기술하고 2장에서는 웹서비스 소비자에서 필요로 하는 기술과 라이브러리 사용법을 설명한다. 3장의 웹서비스 제공자에서는 일반 웹페이지를 분석, 변환하는 파싱 웹서비스와 음성인식 웹서비스를 소개하며 4장에서는 실험 및 결과를 기술한다. 마지막으로 5장에서 결론을 기술하는 구성이다.

## II. 웹서비스 소비자

본 연구에서는 두 개의 사용자를 포함하고 있다. 첫 번째는 사용자를 웹서비스를 사용하는 웹서비스 소비자(Web Services Consumer)이고 두 번째는 웹사이트 기능을 포함한 웹서버를 이용하는 웹사이트 사용자(Web Site Consumer)이다. 본 연구에서 웹서비스 소비자는 파싱 웹서비스 제공자와 음성인식 웹서비스 소비자를 호출하는 기능을 포함한 웹사이트 서버이다.

### 1. 웹서비스 호출

웹서비스에 접근하는 방법은 HTTP-GET 또는 HTTP-POST를 경유하거나 SOAP 메소드를 경유하는 것이다. SOAP는 웹서비스의 인터페이스를 정의하는 WSDL(Web Service Discription Language)을 이용하여 웹서비스를 이

용하는 아주 직접적인 방법이다. .NET 컴파일러는 웹 메소드에 접속하기 위한 세세한 것을 캡슐화하는 특별한 프록시 클래스를 만든다. 프록시 클래스는 클라이언트와 웹서비스 사이의 모든 통신을 조절하며, HTTP에 대한 SOAP Request를 패키징하고, 애플리케이션에 의해 기대되는 반환 데이터 타입에 대해 SOAP으로 인코딩된 응답을 정리한다[9].

### 2. 프록시 라이브러리

프록시를 생성하는 방법은 VisualStudio.NET Tool을 이용하여 웹서비스를 참조하는 방법과 WSDL 명령을 이용하는 방법으로 구분할 수 있다.

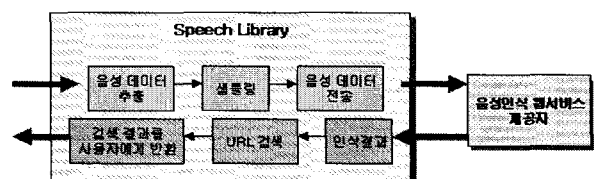
VisualStudio.NET Tool을 이용하여 웹서비스를 사용하기 위해서는 Tool에서 제공하는 '웹참조' 기능을 이용하면 된다. 이와 같은 경우, 음성을 전송하기 위하여 WSE(Web Service Extension)을 설치하고 Microsoft.Web.Service.dll을 참조해야 한다.

본 연구에서는 WSDL 명령을 이용하여 프록시를 생성한 후 컴파일하여 컨트롤을 생성하였다. 컴파일된 컨트롤은 필요한 웹서비스 제공자들을 자동적으로 호출하므로 추가적인 기술 습득 없이 사용할 수 있다.

### 3. 음성 라이브러리

기존의 '음성인식용 웹페이지 변환을 위한 웹서비스 구현'[6]에서는 웹사이트 사용자로부터 녹음된 음성을 DIME(Direct Internet Message Encapsulation) 타입으로 음성인식 웹서비스 서버에게 전송하였다. DIME을 이용하기 위해서는 WSE 설치가 필요하며 또한 전송하여야 하는 음성데이터의 크기는 약 43kByte 정도이다. 본 연구에서는 녹음된 음성에서 필요한 음성데이터를 추출하는 전처리 과정과 음성인식 웹서비스 제공자를 호출하는 과정을 컨트롤로 구현하였다. 전처리 과정을 통해 음성데이터만 추출하면 전송되는 데이터 크기를 10kByte 이하로 감소시킬 수 있다.

본 연구에서는 그림 2와 같은 과정으로 음성 데이터를 추출하고 음성인식 웹서비스 제공자에게 전송한다. 음성인식 웹서비스 제공자에서 인식을 수행한 후 반환되는 결과를 이용하여 해당하는 웹페이지를 검색한 후 웹사이트 사용자에게 반환한다.



▶▶ 그림 5. 음성 라이브러리 구성

### Ⅲ. 웹서비스 제공자

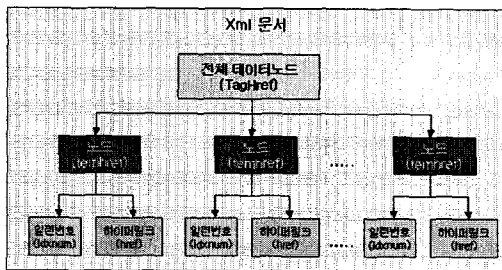
본 연구에서 웹서비스 제공자는 웹페이지를 변환하는 파싱 웹서비스 제공자와 음성인식과정을 수행하는 음성인식 웹서비스 제공자로 구성되어 있다.

#### 1. 파싱 웹서비스 제공자

파싱 웹서비스 제공자는 웹서비스 소비자가 변환하고자 요청하는 일반 웹페이지를 수신하여 태그를 분석하고 변환하고 하이퍼링크를 추출하여 xml로 저장한다.

본 연구에서는 웹페이지의 태그를 분석하여 위하여 정규식을 이용하였다. 정규식은 문자열 내에 포함되어 있는 특별한 조건을 만족하는 문자열을 검색하기 위해 미리 정의된 특수 문자들의 조합을 나타낸다[10]. 정규식을 작성하기 위하여 .NET Framework class library에서 제공하는 Regex 객체와 Match 객체를 사용하였다[6]. 본 연구에서 변환해야 하는 태그는 다음과 같다. 첫 번째 태그 분석은 <body> 태그를 분석한 후 <script...</script>를 추가한다. 두 번째 기능은 하이퍼링크를 포함한 <a href...>.</a>과 이미지버튼을 검색한다. 이미지버튼은 버튼과 같이 이벤트 지향적인 방법으로 사용하므로 하이퍼링크를 사용하지 않는 단점이 있다. 그러므로 웹개발자는 이미지버튼의 alt 속성을 구성해야 한다. 세 번째, ActiveX 컨트롤을 지시하는 <Object...> 태그를 분석한다. ActiveX 컨트롤은 웹개발자가 웹페이지에 미리 추가하고 크기를 0×0으로 지정하여 보이지 않게 하는 경우와 ActiveX 컨트롤을 추가하지 않는 경우를 고려하였다. 또한 웹페이지에 플래시 이미지도 <Object...> 태그로 삽입되므로 ActiveX 컨트롤이 아닌 <Object...> 태그는 변환하지 않아야 한다.

마지막으로 하이퍼링크와 이미지버튼의 alt 속성에 포함된 링크를 xml로 저장한다. 그림 4는 본 연구에서 생성한 링크 테이블 관리자의 구조이다. 링크 테이블 관리자는 xml 문서로 저장된다. 'tmphref' 엘리먼트는 하나의 링크 노드이고 'idxnum' 엘리먼트는 발음할 일련번호, 'href' 엘리먼트는 일련번호와 매핑되는 url 이다[6].



▶▶ 그림 6. 링크 테이블 관리자 구조

### 2. 음성인식 웹서비스 제공자

음성인식 웹서비스 제공자는 기존의 '음성인식을 위한 웹서비스'[3]에서 구현한 시스템을 사용하였다. 이 시스템은 영부터 구까지의 숫자음을 분석하고 인식한다[6].

음성신호분석과정에서는 이미 웹서비스 소비자에서 샘플링된 음성데이터를 수신받아 특징벡터인 캡스트럴계수를 생성한다. 코드북 생성과정에서는 K-means 알고리즘을 이용하여 수신받은 음성데이터와 비교할 수 있는 참조패턴을 저장한다. VQ 인식은 코드북과 음성데이터를 비교하고 인식을 수행한다.

## IV. 실험 및 결과

### 1. 실험

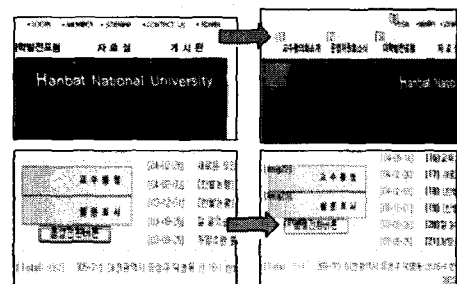
본 연구에서 구현한 시스템을 실험하기 위한 환경은 Windows XP, .Net framework 1.1, Visual Studio.NET Tool을 이용하였다.

실험과정은 일반 웹페이지가 파싱 웹서비스를 이용하여 음성인식이 가능한 웹페이지로 변환하는 단계와 변환된 웹페이지에서 ActiveX 컨트롤을 이용하여 음성을 녹음하고 녹음한 음성과 매핑되는 웹페이지로 전환하는 단계로 진행된다.

실험은 몇 가지의 준비과정을 필요로 한다. 먼저 변환하고자 하는 웹페이지에 파싱 웹서비스를 호출하기 위한 버튼이 필요하다. 버튼은 프록시 라이브러리를 이용하여 파싱 웹서비스를 호출하고 변환된 문서를 수신한 후 파일로 저장한다. 이때 변환하고자 하는 문서의 URL과 수신받은 문서를 저장할 절대 경로를 필요로 한다.

두 번째, 변환된 웹페이지에서 ActiveX 컨트롤로 음성을 녹음한 후 speech 컨트롤을 이용하여 음성데이터를 추출하고 음성인식웹서비스 제공자를 호출하는 'sendDime.aspx' 파일이 필요하다. 이 파일은 본 연구에서 구현한 파일이지만 경우에 따라서는 직접 생성하여도 무관하다.

마지막으로 웹서버에 본 연구에서 구현한 프록시 라이브러리와 음성 라이브러리를 추가하여야 한다.



▶▶ 그림 7. 태그 분석 후 일련번호 삽입



▶▶ 그림 8. ActiveX 컨트롤 삽입

웹페이지 변환 과정의 성능을 알아보기 위하여 현재 사용되고 있는 웹페이지에 음성전환버튼을 추가하였다. 음성전환버튼을 클릭하였을 때 그림 4와 같이 하이퍼링크와 alt 속성을 포함한 이미지버튼에 일련번호가 부여되는 것을 확인할 수 있으며 ActiveX컨트롤 또한 그림 5에서와 같이 삽입된 것을 알 수 있다.

이 단계에서는 웹페이지를 재구성할 뿐만 아니라 링크 테이블을 xml 문서로 생성한다. 이미지버튼의 alt 속성에 포함된 링크와 하이퍼링크, 그리고 메일 이미지에 부여된 하이퍼링크 까지 검색한 것을 확인할 수 있다.

변환된 웹페이지에 부여된 ActiveX 컨트롤은 음성녹음이 종료되면 녹음된 음성에서 음성데이터를 추출한 후 샘플링하여 음성인식 웹서비스 제공자에게 전송한다.

음성녹음은 2초간 진행되며 2초가 경과하면 자동으로 11kHz, 비트샘플링수는 16bit, 녹음형태는 모노로 설정하여 자동으로 저장된다[6]. 실험대상은 화자중속 남성2인, 화자중속 여성2인, 화자독립 남성2인, 화자독립 여성 2인의 음성으로 2회씩 녹음하여 실험하였다. 화자중속적인 음성은 약 85% 이상, 화자독립적인 음성은 약 70% 정도 인식이 되며 인식결과를 해당하는 url로 찾는 결과는 98% 정도로 확인되었다.

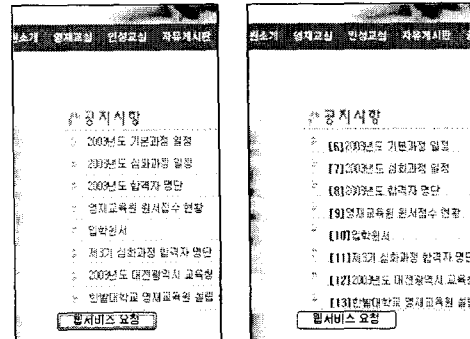
## 2. 검토 사항

본 연구에서 실험으로 이용한 웹페이지에서는 변환이 정확하게 수행되고 링크 테이블도 정확하게 저장되는 것을 확인할 수 있었다. 그러나 다양한 웹사이트와 다양한 웹페이지에서도 적용되는 알아보기 위하여 같은 웹사이트에 존재하는 다른 웹페이지와 한밭대학교 영재교육원 웹사이트에 존재하는 웹페이지로 재실험 하였다.

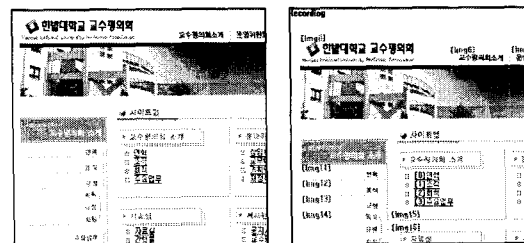
그림 6과 그림 7을 통하여 정확하게 태그를 분석하고 일련번호와 ActiveX 컨트롤을 삽입하는 것을 확인하였다. 그림 8은 웹페이지에서 추출된 링크 테이블의 일부이다.

이와 같은 실험으로 파싱 웹서비스 제공자를 이용하여 일반 웹페이지를 음성인식용 웹페이지로 변환이 정확하게 이루어지는 것을 알 수 있다. 또한 음성인식 웹서비스 제공자에서 녹음

된 음성을 잘못 인식하는 경우를 제외하고는 해당하는 URL을 정확히 검색하고 반환하는 것도 알 수 있다.



▶▶ 그림 9. 다른 웹사이트의 웹페이지 변환



▶▶ 그림 10. 같은 웹사이트의 다른 웹페이지 변환

```

<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<!DOCTYPE TempTable [view Source for full doctype...]>
<!-- 이 파일은 see태그의 임시저장소입니다 -->
<Taghref>
- <tmhref>
  <idnum>0</idnum>
  <href>introduce/intro.aspx?&#38;biname=intro</href>
</tmhref>
- <tmhref>
  <idnum>1</idnum>
  <href>gifted/math_class.aspx?&#38;biname=math_class</href>
</tmhref>
+ <tmhref>
  <idnum>2</idnum>
  <href>human/human class.aspx?&#38;biname=human class</href>
</tmhref>

<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<!DOCTYPE TempTable [view Source for full doctype...]>
<!-- 이 파일은 see태그의 임시저장소입니다 -->
<Taghref>
- <tmhref>
  <idnum>1</idnum>
  <href>../about/history.aspx?&#38;folder=about&#38;left=history</href>
</tmhref>
- <tmhref>
  <idnum>2</idnum>
  <href>../about/organization.aspx?&#38;folder=about&#38;left=orga
  
```

▶▶ 그림 11. 두 개의 웹페이지에서 추출된 링크 테이블

## V. 결론

본 연구에서는 별도의 음성인식시스템을 구축하거나 음성데이터베이스를 구축하지 않고 일반 웹페이지에서 음성인식 시스템을 사용하는 방안을 제안한다.

웹서비스로 제공되는 음성인식 시스템을 사용하기 위해서는 첫 번째로 일반 웹페이지가 음성인식이 가능한 웹페이지로 변

환되어야 한다. 두 번째로 WSDL을 이용하여 필요한 웹서비스의 위치를 파악한 후 웹서비스를 참조하여야 한다. 마지막으로 녹음된 음성데이터를 DIME 타입으로 전송해야 한다.

본 연구에서는 이와 같은 기능을 프록시 라이브러리와 음성 라이브러리로 구현함으로써 웹사이트 개발자가 음성인식 기술이나 웹서비스에 관한 기술을 습득하지 않아도 음성인식 웹서비스를 사용할 수 있도록 제공한다. 또한 일반 웹페이지를 음성인식이 가능한 웹페이지로 전환할 수 있는 파싱 웹서비스 제공자를 구현하였다.

구현된 시스템의 성능을 알아보기 위하여 두 개의 웹사이트에서 3장의 웹페이지로 실험하였다. 실험 결과, 프록시 라이브러리가 파싱 웹페이지를 호출하고 변환된 웹페이지를 저장하는 것을 알 수 있었다. 음성 라이브러리는 녹음된 음성을 샘플링한 후 음성인식 웹서비스 제공자에게 전송하며 음성인식 웹서비스 제공자에서 인식된 인식결과에 해당하는 웹페이지를 검색하여 사용자에게 반환하는 것을 확인하였다.

이 연구를 토대로 하여 다양한 언어로 개발된 웹사이트에 적용될 수 있는 컨트롤과 일련번호가 아닌 특정 단어를 인식할 수 있는 인식시스템을 개발 중에 있다.

#### ■ 참고 문헌 ■

- [1] 이항섭외, "웹 브라우저 상에서 한국어 음성인식을 이용한 정보검색 시스템", 음성통신 및 신호처리 워크샵, 15권1호, 1998, pp.35-38.
- [2] 장영건의, "한국어 음성 웹브라우저 설계 및 구현", 정보과학회논문지, 제7권5호, 2001.10, pp.458-466.
- [3] 오신영, "웹서비스를 위한 음성인식시스템의 구현", 한밭대학교, 정보통신전문대학원 논문집, 제1권1호, 2003.2, pp.87-97.
- [4] 오지영외2인, "웹환경에 기반한 음성인식시스템의 구현" 한국기술응용학회, 2002 춘계학술발표논문집, 2002.6, pp.70-76.
- [5] 오지영, "XML 웹서비스와 인식기술을 이용한 웹브라우저", 한밭대학교, 정보통신전문대학원 논문집, 제1권1호, 2003.2, pp.99-114.
- [6] 오지영, "음성인식용 웹페이지 변환을 위한 웹서비스 구현", 한밭대학교, 멀티미디어학회논문지, 제7권8호, 2004.8, pp.1162-1169.
- [7] 이상엽, "Visual C++ Programming Bible", 영진출판사, 1998.
- [8] 김용성, "Visual C++ Programming Bible", 영진출판사, 2002.
- [9] Ashish Banerjee, C# Web Services, 2002, Wrox Press.
- [10] 정규식(regular expression)의 정의와 특수문자 정의, "http://cne.ks.ac.kr/~hgkim/ce315/php3/regular.html", 2003.