

서버 기반 웹 리더 *kWebAnywhere*의 설계 및 구현

Design and Implementation of Server-Based Web Reader *kWebAnywhere*

윤 영 선¹⁾
Yun, Young-Sun

ABSTRACT

This paper describes the design and implementation of the *kWebAnywhere* system based on *WebAnywhere*, which assists people with severely diminished eye sight and the blind people to access Internet information through Web interfaces. The *WebAnywhere* is a server-based web reader which reads aloud the web contents using TTS(text-to-speech) technology on the Internet without installing any software on the client's system. The system can be used in general web browsers using a built-in audio function, for blind users who are unable to afford to use a screen reader and for web developers to design web accessibility. However, the *WebAnywhere* is limited to supporting only a single language and cannot be applied to Korean web contents directly. Thus, in this paper, we modified the *WebAnywhere* to serve multiple language contents written in both English and Korean texts. The modified *WebAnywhere* system is called *kWebAnywhere* to differentiate it with the original system. The *kWebAnywhere* system is modified to support the Korean TTS system, VoiceText™, and to include user interface to control the parameters of the TTS system. Because the VoiceText™ system does not support the *Festival* API used in the *WebAnywhere*, we developed the *Festival* Wrapper to transform the VoiceText™'s private APIs to the *Festival* APIs in order to communicate with the *WebAnywhere* engine. We expect that the developed system can help people with severely diminished eye sight and the blind people to access the internet contents easily.

Keywords: Server-based Web Reader, Web Accessibility, Text-To-Speech, *kWebAnywhere*

1. 서론

최근 UN의 발표에 따르면 전 세계 인구 중에서 장애를 가진 인구의 비율이 10%를 차지한다고 한다[1]. 미국과 유럽의 경우에 비슷한 비율을 보이고 있으며, 2012년에 발간된 미국의 센서스 보고에 의하면 2010년 현재 전체 인구의 19%가 장애인으로 파악되며[2], 유럽의 경우에는 15%에 육박하는 것으로 나타나고 있다. 한국의 경우 등록된 장애인이 5%로 집계되나[3], 보건복지부에 등록되지 않거나 정서상 숨겨진 장애인

을 포함한다면 그 비율은 유엔의 통계와 비슷할 것으로 예측된다. 이들 통계에서 전체 장애인 중 약 10%가 시각 장애인으로 집계되고 있어, 전체 인구의 약 0.5~1.0%를 시각 장애인으로 볼 수 있다.

최근 복지 정책에 대한 논의가 활발히 진행되면서 노약자와 장애인들에 대한 사회적 관심이 증가하고 있으며, 접근성(accessibility)에 대한 많은 연구가 진행되고 있다. 접근성이란 사용자의 신체적 특성이나 지역, 성별, 나이, 지식수준 등의 제한 사항을 고려하여 가능한 많은 사용자가 불편 없이 제품이나 서비스를 이용할 수 있도록 하는 것을 말한다[4]. 접근성이 높다는 것은 제약 사항을 가진 사용자도 불편 없이 서비스나 제품을 사용할 수 있다는 것을 말하며, 접근성이 낮다는 것은 관련 서비스나 상품을 사용하기 불편하거나 사용할 수 없다는 것을 말한다. 접근성에 대한 배려가 부족한 최근의 모바일 기기나 컴퓨터의 급속한 발전은 노약자나 장애인들이 서비스나 제품을 사용하기 어렵게 만드는 접근성 갭(accessibility

1) 한남대학교, ysyun@hannam.kr, 교신저자

본 연구는 지식경제부 및 한국산업기술평가관리원의 QoLT 기술혁신사업[10036459, QoLT산업기술기반 지원센터 구축사업]의 일환으로 수행되었음.

접수일자: 2013년 7월 13일

수정일자: 2013년 9월 27일

게재결정: 2013년 11월 14일

gap)이라는 문제를 발생시키고 있다[5].

인터넷 및 모바일 환경이 발달되고 다양한 형태의 정보가 유통되면서 장애인과 노약자들을 위한 접근성 문제는 개인의 문제가 아닌 사회적 문제로 인식되고 있다. 접근성이 충분히 고려되지 않은 컴퓨터 환경은 장애인이나 노약자들의 사회적 고립 현상을 야기하기도 한다. 최근에는 접근성이 고려된 스마트폰이나 태블릿의 확산이 증가하고 있으나, 접근성이 고려된 기기의 활용도는 여전히 저조한 편이며, 아직도 많은 사람들은 개인이나 공공시설의 데스크톱 컴퓨터를 이용하여 정보에 접근하고 있다.

컴퓨터 사용을 지원하기 위하여 여러 보조 지원 장치나 소프트웨어가 사용되고 있으나 비용이나 작동 환경, 장애 유형에 따라 모든 사용자가 보조 지원 장치를 사용할 수 있는 것은 아니다. 사용자들은 장애 유형이나 정도에 따라 키보드나 마우스와 같은 입력 장치, 스크린 리더나 화면 확대경, 점자 출력 장치와 같은 출력 장치 등의 지원을 받는다. 본 연구는 여러 장애 유형들 중에서 저시력자나 전맹인과 같이 시력이 불편한 사용자들이 웹 정보에 접근하여 정보의 공유 등에 참여할 수 있도록 클라이언트 컴퓨터에 별도의 소프트웨어 설치 과정 없이 접근성 기능을 지원하는 서버 기반의 웹 출력 지원에 관한 연구이다.

본 논문의 구성은 다음과 같다. 2장에서는 접근성의 소개와 접근성 지원 도구에 대하여 설명하고, 3장에서는 기존 *WebAnywhere*의 구성 및 한계를, 4장에서는 본 연구에서 구현한 *kWebAnywhere*의 설계 개념 및 구성, 구현 방식에 대하여 기술한다. 5장에서 본 연구에 대한 요약 및 결론으로 끝을 맺는다.

2. 연구 배경

접근성은 목적 및 환경에 따라 생활(이동) 접근성, 컴퓨터 접근성, 웹 접근성, 모바일 접근성 등 다양하게 분류되고 있다. 일반적으로 생활 접근성은 장애인이나 노약자들이 보조기기를 이용하여 이동이나 생활에 불편함이 없도록 지원하는 활동이나 수준을 말한다. 컴퓨터 접근성은 컴퓨터를 쉽고 효율적으로 사용할 수 있도록 시각 장애인용 점자 터미널, 넓은 활자의 키보드, 특수하게 제작된 마우스 등과 같은 장치를 이용하거나 화면에 표시된 내용을 음성으로 변환하여 읽어주는 스크린 리더(screen reader) 소프트웨어 등에 의하여 컴퓨터를 사용할 수 있는 수준이나 환경을 정의한 것이다. 웹 접근성은 장애를 가진 사람들의 웹 정보에 대한 접근 정도를 나타내는 것으로서 홈페이지의 디자인 및 구축 등에 관하여 정의하고 평가한다[6]. 최근에는 스마트폰이나 태블릿과 같은 모바일 단말기 등의 발달에 따른 모바일 환경에서의 접근 용이성을 정의한 모바일 접근성도 활발히 연구되고 있다[7]. 이 중에서 본

연구는 웹 접근성에 초점을 맞추고 있다.

인터넷의 발달과 더불어 소개된 웹 시스템(WWW; World Wide Web)은, 분산되어 있는 정보를 하이퍼링크(hyperlink)를 통하여 효과적으로 표현하고 접근할 수 있어 인터넷 트래픽의 대다수를 차지하고 있다. 웹 시스템의 창시자인 팀 버너스리(Tim Berners-Lee)는 “The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.”[8]라고 접근성의 기본 개념을 언급하였으며, WWW 컨소시엄(W3C)에서는 웹은 기본적으로 물리적, 정신적 능력과 환경, 언어, 문화, 위치에 무관하게 모든 사람들에게 동작할 수 있도록 설계되어야 한다고 주장하고 있다[9]. 웹 사이트가 올바르게 설계되어 개발되고 편집될 때 모든 사용자들은 정보와 기능에 동등하게 접근할 수 있으며, 사이트에 기여할 수 있기 때문이다[6]. 이러한 기본 원리에 바탕을 둔 웹 접근성은 <표 1>의 장애들을 고려하고 있다.

표 1. 장애의 종류와 증상(내용)[10]
Table 1. Types of disabilities and their symptoms[10]

장 애	설 명
시 각	시력, 색깔 이상, 다양한 형태의 저시력을 포함한 시각 장애
청 각	난청이나 특정 유형의 소리에 민감하게 반응하는 상태
이동성	파킨슨 병, 근육병, 뇌졸중마비, 뇌졸중과 같은 조건으로 인한 근육 속도 저하, 근육 제어 손실로 말미암아 손을 쓰기 어렵거나 쓸 수 없는 상태
발 작	감박이는 효과나 시각적인 스트로보스코프를 통해 일어나는 간질성 발작
인 지	문제 해결과 논리 능력, 집중력, 기억력에 문제가 있는 정신 지체 및 발달 장애, 학습 장애(난독증, 난산증 등)

<표 1>에서 살펴본 바와 같이, 장애의 종류에 따라 그 반응 정도가 다르기 때문에 모든 장애를 고려한 접근성은 특정 학문이나 분야가 아닌 다양한 학문의 협조로 연구되어야 한다. 본 연구에서는 시각 장애인들이 인터넷 환경에서 웹을 사용하고자 할 때 별도의 소프트웨어나 장치의 설치 없이 전용 환경과 비슷한 지원을 받을 수 있는 지원 방법을 고려하였다.

웹 접근성을 지원하기 위한 특수한 형태로 텍스트를 음성으로 변환(TTS; Text-to-Speech Synthesis)하여 사용자가 쉽게 내용을 확인할 수 있도록 하는 리더(reader)가 많이 사용된다. 일반적으로 스크린 리더는 컴퓨터 화면에 표시되는 거의 모든 텍스트 출력을 음성으로 변환하여 읽어주는 소프트웨어를 말하며, 웹 리더(web reader)는 웹 브라우저를 통하여 인터넷에 접속하였을 경우에만 해당되는 페이지를 음성으로 변환하여

출력하는 것을 말한다. 따라서 스크린 리더의 경우 운영체제에 영향을 받으며, 웹 리더의 경우 구현 방법에 따라 기능의 제한을 받는다.

기존의 웹 접근성 지원 연구는 구현 방식에 따라 크게 스크린 리더 기반, 클라이언트(웹 브라우저) 기반, 서버 기반으로 구분할 수 있다. 스크린 리더 기반의 경우 특정 운영체제에서 동작하는 스크린 리더 기능을 이용하여 웹 정보에 접근하는 방법으로서 윈도우즈 환경의 JAWS[11], 센스리더(sense reader)[12], WindowsEyes[13] 등이 있으며, 맥 환경의 VoiceOver 등이 있다. 클라이언트 기반 지원 방안은 기존 브라우저의 확장 또는 플러그인 형태로 동작하며 FireVox[14], ChromVox[15]등이나 독립 프로그램으로서 독자적인 브라우저[16],[17] 기능을 제공한다. 마지막으로 서버 기반의 구현 방법은 각 브라우저에서 지원하는 운영체제의 표준 오디오 출력 형식이나 널리 사용되는 어도비 플래시 방식의 음성 출력 형식을 선택하여 운영체제나 특정 브라우저에 독립적으로 구현한 방식으로서 *WebAnywhere*[18]와 *TextReader WAS*[19] 등이 있다(<표 2> 참조).

이러한 방식들 중에서 공공 기관이나 외부에서 저시력자나 전맹인들이 웹 정보에 쉽게 접근할 수 있도록, 별도의 소프트웨어 설치 없이 웹 접근성을 지원하는 방법으로 서버 기반의 웹 리더 구현 방식을 고려하였다.

본 연구에서는 서버기반의 웹 접근성을 지원하기 위한 방안으로 워싱턴 주립대학에서 연구된 *WebAnywhere*[18],[20]를 채택하였으며, 센스리더에 채택된 (주)보이스웨어의 음성합성기[19]를 적용하여 한국어 음성을 출력하게 하였다. 따라서 본 연구에서 채택한 음성 합성기의 음질이나 성능은 기존 센스리더와 동일하기 때문에 별도의 음성 합성기에 대한 성능 평가를 진행하지 않았다. 처음 공개된 *WebAnywhere*는 영어 합성기를 이용하며, 한국어를 포함한 다중언어 음성을 고려하지 않았으나 계속된 버전 업데이트를 통하여 언어 환경(locale)에 따라 언어별로 음성 합성기를 선택할 수 있도록 하였다. 그러나 영어와 한글이 혼재된 홈페이지와 같은 환경에서는 기존의 *WebAnywhere*를 적용하는 것이 어려워 한국어 음성 합성기를 적용한 *kWebAnywhere*를 설계 및 구현하게 되었다.

3. WebAnywhere의 소개

본 장에서는 기존에 발표된 문헌[18],[20]을 바탕으로 *WebAnywhere*의 설계 및 구성, 한계를 살펴본다.

3.1 설계 및 구성

*WebAnywhere*는 저시력자나 전맹인과 같은 시각장애인들이 손쉽게 웹 정보를 접근할 수 있도록 웹 페이지의 문서 객체 모델(DOM; Document Object Model)에 직접적으로 상호 작용하도록 설계되었다. *WebAnywhere*는 로드된 웹 페이지의 DOM 객체에 접근하기 위하여 사전에 순서가 부여된 깊이 우선 탐색(depth first search)를 이용한다. 깊이 우선 탐색의 순서는 화면에서 왼쪽에서 오른쪽으로 문서를 읽어가는 것처럼 순서가 부여되어 있다. 각각 DOM 원소에 도달하면 해당되는 원소의 속성에 맞게 음성으로 출력한다. 일반 텍스트의 경우에는 직접 음성 출력을 호출하며, HTML 태그 명령을 만나는 경우에는 해당 태그를 적절한 형태로 변환하여 음성을 출력한다. 사용자는 키를 이용하여 DOM 원소간의 이동이 가능하며, 이 이동을 통하여 문장이나 단어의 음성 출력을 건너뛸 수 있다.

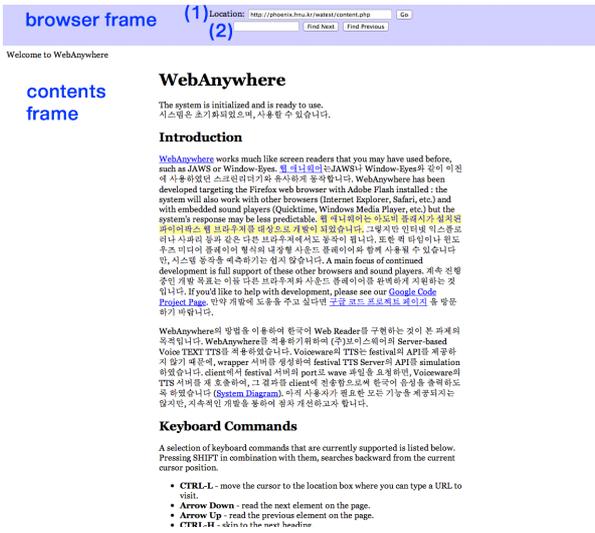
*WebAnywhere*의 화면은 자체적으로 URL을 입력하여 이동할 수 있는 주소 입력 프레임(browser frame)과 웹 페이지를 출력하는 내용 프레임(content frame)으로 구성되어 있다. 사용자는 웹 페이지에서 마우스나 키보드를 이용하여 링크된 페이지로 이동할 수 있거나, (1) 직접 주소 입력 프레임(browser frame)의 location 필드에 URL을 입력하여 이동할 수 있다(<그림 1> 참조).

표 2. 웹 접근성 방식의 구현 방법의 비교
Table 2. Comparison of web accessibility implementation methods

구현 방법	장점	단점
스크린 리더	웹 페이지 내용뿐만 아니라, 메일, 편집 등 거의 모든 데스크톱의 화면 출력에 대하여 음성 지원을 제공	특정 운영체제에서만 동작하기 때문에, 플랫폼이 변하는 경우 적용하기 힘들 프로그램의 구매 비용(영어 약 \$800~\$1000, 한국어 65만 원 정도)이 소요 관리자 권한으로 프로그램의 설치가 필요
브라우저 확장 웹 리더	상대적으로 운영체제 환경에 덜 종속적 무료	웹 정보만 지원 웹 정보를 보기 위해서는 특정 웹 브라우저만을 사용 사용자가 확장 기능을 추가로 설치 동적 페이지 등과 같이 스크립트나 복잡한 페이지의 경우 오동작이 우려
서버 기반 웹 리더	프로그램의 설치가 필요하지 않음 무료 공공장소와 같이 외부의 컴퓨터에서 쉽게 사용 가능 컴퓨터의 운영체제나 웹 브라우저에 상관없이 쉽게 사용 가능	웹 정보만 지원 동적 페이지 등과 같이 스크립트나 복잡한 페이지의 경우 오동작이 우려 flash를 지원하지 않는 iPhone이나 iPad와 같은 모바일의 경우, 지원이 어려울 수 있음

표 3. WebAnywhere 시스템 모듈의 구성
Table 3. System module structure of WebAnywhere

모듈 구성	기능
클라이언트 스크립트	사용자 상호 작용 지원, 음성 출력 지원
서버 음성 합성 시스템	음성 합성, 캐시 지원
서버 프락시	교차 서버 스크립트 지원, 웹 페이지 캐시



(a) 전체 구성도 (Overall layout)



(b) browser frame 확대 (Magnification of browser frame)

그림 1. WebAnywhere 구성도[18], [20].
Figure 1. Layout of WebAnywhere[18], [20].

WebAnywhere는 인터넷 접근과 음성 출력을 위하여 대부분의 컴퓨터에서 기능이 동작하도록 설계되었다. 따라서 특정 브라우저나 플러그인에 독립되게 구성되었으며, 클라이언트에 특정 프로그램이나 플러그인을 설치하지 않으며, 서버에서 프로그램을 다운로드 받거나 서버로 업로드 하지 않는다. 또한, 클라이언트에 따라 음성을 출력하는 기본 프로토콜이 다르기 때문에 공통된 기술을 적용하기 위하여 플래시 기반의 음성 출력 기법을 적용하였으며, 서로 다른 브라우저와 운영체제에서 음성 출력이 가능하도록 다양한 음성 출력을 지원한다.

WebAnywhere는 클라이언트 측 스크립트, 서버 측 음성 합성 시스템 및 캐시, 서버 측 프락시의 세 모듈로 구성되어 있다. 클라이언트 측에서는 사용자 상호작용 및 음성 출력[21]에 관한 부분을 담당한다. 서버 측의 음성 합성 시스템은 입력된 텍스트를 음성으로 출력한 후, WebAnywhere가 설치되어 있는 서버에 음성을 저장하여 음성 합성 시스템이 다른 시스템에 설치되어 있더라도 음성 출력을 빠르게 지원하도록 하였다. WebAnywhere에서 이용하는 음성 합성기는 Festival 음성 합성기[22]를 사용하였다. 마지막으로 서버 측의 프락시[23]는 웹 페이지에 내재된 스크립트가 서로 다른 서버에 존재하는 경우 교차 서버의 제한으로 인하여 동작되지 않는 현상(cross-site scripting restriction)을 해결하기 위하여, 원래 서버의 웹 페이지를 WebAnywhere가 설치된 서버에서 서비스하는 것처럼 변환하여 웹 서비스를 지원한다(<표 3> 참조).

3.2 한계

현재 공개된 WebAnywhere 시스템은 어도비 플래시를 지원하지 않는 웹 브라우저의 경우 효율적인 지원이 어렵다. WebAnywhere는 플래시 기능을 이용하여 단어나 문장단위의 이동이나 음성 입출력의 시작 및 정지 기능을 구현하였기 때문이다. 만약 운영체제의 기본 음성 입출력 기능(예; 윈도우즈 환경의 Media Player, 맥 환경의 Quick Time Player 등)을 이용할 경우 사용자 대화식이 아닌 웹 페이지의 연속 읽기만이 지원된다.

특정 운영체제에 설치되는 스크린 리더의 경우 음성 출력 과정에서 음성 인식 기능을 이용한 명령어 입력이 가능하나, WebAnywhere의 경우 서버 기반의 교차 스크립트 구현 방식이기 때문에 클라이언트의 지원 기능을 이용할 수 없다는 단점이 존재한다.

또한, 웹 프락시 기능을 이용하여 원래의 웹 페이지를 WebAnywhere가 설치된 서버로 다운로드한 후, 다시 서비스하는 방식이기 때문에 원 문서를 제공하는 시스템의 보안 정책이 안정적으로 동작하기 어려울 수 있다. 즉, 웹 프락시는 교차 서버 스크립트의 문제를 해결하기 위하여 웹 문서를 다운로드 한 후 다시 서비스하고 있기 때문에, 다운로드 받은 문서의 보안 정책이나 프로그램이 훼손될 수 있다.

마지막으로 시스템에서 제공하는 스크린 리더의 경우 화면의 출력을 음성으로 변환하기 때문에 화면에 표시되는 모든 텍스트를 음성으로 출력 가능하다. WebAnywhere의 경우 브라우저의 플러그인에 의하여 표시되는 pdf 파일이나 다른 3자 제공의 파일 형식의 경우(Microsoft사의 doc이나 ppt, xls 파일 편집 또는 보기 기능)에 브라우저의 기능과는 독립적이기 때문에 음성 출력이 어렵다는 한계가 존재한다.

4. kWebAnywhere의 설계 및 구현

본 장에서는 한국어 지원을 위하여 수정된 WebAnywhere 시스템의 설계와 구성을 살펴보고, 구현 과정을 기술한다. 또한 현재 구성된 시스템의 추후 구현 및 확장에 대하여 검토한다.

4.1 설계 및 구성

한국어 지원 *WebAnywhere*(이하 *kWebAnywhere*²⁾)를 구현하기 위하여 기존의 *WebAnywhere* 시스템에 한국어 음성 합성기 [19]를 이용하도록 수정하였으며, 음성 합성기에 파라미터를 전달할 수 있도록 *dhtmlx*[24] 라이브러리를 채택하였다. *dhtmlx* 라이브러리를 이용한 도구 바 또는 도구 박스는 쿠키(cookie) 정보를 이용하여 클라이언트별로 설정사항을 저장할 수 있도록 하였다. 한국어 음성 합성기는 자체 API(Application Program Interface)를 채택하고 있기 때문에, *WebAnywhere*와 통신할 수 있도록 변환기(wrapper)를 구현하여 *Festival* 음성 합성기와 호환하도록 하였다(<그림 2> 참조).

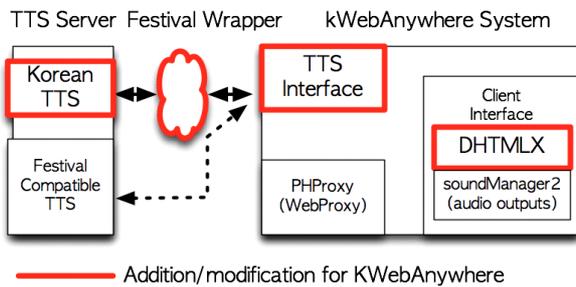


그림 2. *kWebAnywhere*의 시스템 구성도

Figure 2. System diagram structure of *kWebAnywhere*

기존의 *WebAnywhere* 시스템에서 추가되거나 변경된 구성 요소는 다음과 같다.

1) TTS 음성 합성기(Text-To-Speech Synthesizer)

입력된 한글 및 다국어 텍스트를 음성으로 변환하도록 보이스웨어의 *VoiceText Web Reader*TM를 채택하였다. *VoiceText*TM는 Linux 상에서 동작하는 서버/클라이언트 시스템으로서 32/64비트를 지원하며, API 호출에 따라 파일, 버퍼, 연속 출력 등이 가능하며, 발성 속도, 피치(pitch) 조절, 볼륨(volume) 등을 조절할 수 있다. 그러나 *Festival* API와 호환이 되지 않아 별도의 변환 작업이 필요하다. 주된 기능으로는 한국어, 영어, 중국어, 스페인어가 텍스트로 전달되면 16bit Microsoft Wave PCM 형식으로 파일이 생성되거나 버퍼로 전달된다.

2) *Festival* 변환기

한국어 TTS 합성기와 *WebAnywhere* 시스템이 원활하게 정보를 주고받을 수 있도록 기존의 *WebAnywhere* 시스템에서 *Festival* API로 전달되는 정보를 받아, *VoiceText*TM API로 변환하고 한글 인코딩을 변환하여 *VoiceText*TM TTS 서버를 호

출한다.

3) TTS 인터페이스

WebAnywhere 시스템은 한국어를 고려하지 않고, 한국어 TTS 서버의 출력 형식이 기존 *Festival* 호환 서버와 다르기 때문에, 내부에서 사용되는 한국어 코딩 형식을 네트워크 통신 상에서 손실되지 않도록 UCS2³⁾로 변환한 후 *Festival* 변환기로 전달하고 음성 합성기의 결과를 받아 *kWebAnywhere*가 실행 중인 서버에 mp3 형식으로 저장한다.

4) 클라이언트(사용자) 인터페이스

사용자가 음성 합성기의 발성 속도, 크기, 피치 등을 조절하고, 단축키 또는 버튼 누름에 의하여 음성 합성을 on/off 할 수 있도록 인터페이스를 제공한다.

4.2 구현

본 절에서는 4.1절에서 설명된 각 구성요소의 구현 과정을 설명한다. *VoiceText*TM 합성기는 보이스웨어 시스템의 구성을 그대로 사용하였기 때문에, 합성기를 제외한 나머지 구성 요소의 구현 과정을 설명한다.

1) *Festival* 변환기

Festival 음성 합성기[25]는 다양한 API를 제공하고 있으며, 대표적인 지원 API로는 Lisp 언어와 비슷한 Scheme API, Shell API, 서버/클라이언트 API, C/C++ API, C Only API, Java and JSAPI 등이 있다. 본 연구에서는 서버/클라이언트 네트워크 환경에서 음성 합성을 지원하기 때문에 서버/클라이언트 API를 이용하였다. 한국어 음성 합성기인 *VoiceText*TM가 *Festival* 음성 합성기와 호환되기 위해서는 클라이언트 접속을 지원하고, 클라이언트 API에 호출하는 기능을 *Festival* 변환기에서 지원하여야 한다. 기존 *Festival* 클라이언트 접속 프로토콜의 명령 전달 순서는 <표 4>와 같고, *kWebAnywhere*에서 지원하는 클라이언트 호출 기능은 <표 5>와 같다.

클라이언트에서는 영어와 한글 텍스트를 구분 없이 작은 문장 단위로 서버로 전송한다. *VoiceText*TM 합성기는 다국어 합성기를 지원하나 선택된 합성 엔진에 따라 한국어/영어가 발생된다. 한국어 합성 엔진이 선택된 경우 영어 문장이 입력되면 영어를 한국어로 변환하여 발생하기 때문에 자연성(naturalness)이 현저히 저하된다. 예를 들어 한국어 음성 합성 엔진의 경우 영어, 한글, 스페인어, 중국어 등 4개 국어를 모두 입력으로 받아 음성을 출력할 수 있다. 따라서 *Festival* 변환기에서는 입력 문장을 분석하여 한국어와 영어가 혼용된 경우 한국어 합성 엔진을, 영어만 전용으로 전달된 경우 영어

2) 본 연구에서 구현된 *kWebAnywhere* 시스템은 http://phoenix.hnu.kr/wa_beta/에서 확인할 수 있다.

3) UCS2 : 2-byte Universal Character Set을 의미하며, 모든 문자를 고정된 길이의 16비트로 표현하는 방식

합성 엔진을 사용하도록 하였다.

표 4. Festival 음성합성기 서버/클라이언트 프로토콜
Table 4. Server/client protocol of Festival synthesizer

	서버	클라이언트
클라이언트 송신		① connect to server ② send <i>command</i> (<i>let(utt(Utterance Text "text"))</i> (<i>begin(utt.synth utt)</i> (<i>utt.wave.resample utt 22500)</i> (<i>utt.send.wave.client utt))))</i>
서버 수신	③ receive & parse <i>command</i>	
서버 전송	④ send "WV" ⑤ write wave ⑥ send "LP" ⑦ write checksum ⑧ send "OK"	
클라이언트 수신		⑨ get wave ⑩ save wave for proxy

표 5. kWebAnywhere에서 지원하는 Festival 명령
Table 5. Festival commands supported in kWebAnywhere

클라이언트 명령	설명	비고
Utterance Text	합성기에 전달되는 텍스트	구현
wave.resample	합성기의 출력 샘플링 지정 ⁴⁾	무시
wave.rescale	출력 음성의 볼륨	구현
wave.speed	출력 음성의 발성 속도	추가
wave.pitch	출력 음성의 피치	추가

2) TTS 인터페이스

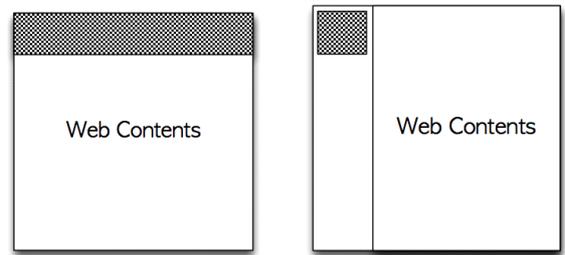
kWebAnywhere는 서버 스크립트와 클라이언트 스크립트, TTS 인터페이스 스크립트로 구성된다. 서버 스크립트는 php, 클라이언트 스크립트는 자바 스크립트로 구현되었다. TTS 인터페이스는 Festival 음성 합성기와 네트워크 통신을 위하여 cgi(common gateway interface)로 구현되었으며, 널리 사용되는 perl 언어로 작성되었다. VoiceText™ 합성기에서 사용하는 한국어 코드와 perl에서 자체적으로 사용하는 한국어 코드가 다르기 때문에 Festival 변환기에서 코드 변환이 이뤄진다. 그러나 한국어와 같이 8비트 문자열의 경우 php 스크립트, 자바스

4) kWebAnywhere은 음성 합성 시스템의 음성 출력 파일에 대해 직접 샘플링 율(sampling rate)을 변환한다.

크립트, perl 스크립트에서 문자열을 처리하는 과정에서 원하지 않는 결과를 가져올 수 있다. 따라서 8비트 문자열의 경우 Escape/Unescape⁵⁾(HTML 코드 변환/텍스트 복원) 과정을 통하여 특수 문자와 8비트 문자가 배제된 문자열로 변환하여 오동작이 발생하지 않도록 하였다. 즉, 내부적으로 한글의 경우 Escape 문자열로 변환되어 사용하도록 하였다.

3) 클라이언트(사용자) 인터페이스

사용자가 클라이언트(브라우저)에서 서버에서 동작하는 음성 합성기의 출력 환경을 제어하기 위하여 지원하는 인터페이스이다. 고려된 방식은 도구 바 형식과 도구 상자 형식이 검토되었으며, 도구 바 형식은 브라우저의 최상단에 위치하고, 도구 상자는 웹 문서의 왼쪽이나 오른쪽에서 문서의 흐름에 따라 이동(floating)할 수 있다(<그림 3> 참조).



(a) 도구 바 (toolbar) (b) 도구 상자 (toolbox)

그림 3. 음성 출력 형식을 제어하는 (a) 도구 바와 (b) 도구 상자의 설계

Figure 3. Layouts of (a) toolbar and (b) toolbox that control speech output format

두 가지 구현 방식 중 도구 상자는 문서의 흐름에 따라 유동적이며, 웹 문서의 왼쪽이나 오른쪽에 위치하기 때문에 많은 사이트에서 채택하는 네비게이터(navigator)와 혼동 가능성이 있어, 최종적으로 도구 바를 채택하여 구현하였다. 도구 바의 구성 버튼과 그 기능은 <표 6>에 정리하였다.

kWebAnywhere에 적용된 도구 바의 버튼 구성은 음성 출력 토글(온/오프) 버튼, 볼륨 조절 슬라이더, 발성 속도 슬라이더, 피치 조절 슬라이더, 초기화 버튼으로 구성되었다. 음성 출력의 경우 키보드 인터페이스를 병행하였으며, 발성 속도와 피치 조절의 경우 현재 마우스에 의하여 조작이 되나 사용자들의 의견을 받아들여 키를 병행하고자 한다.

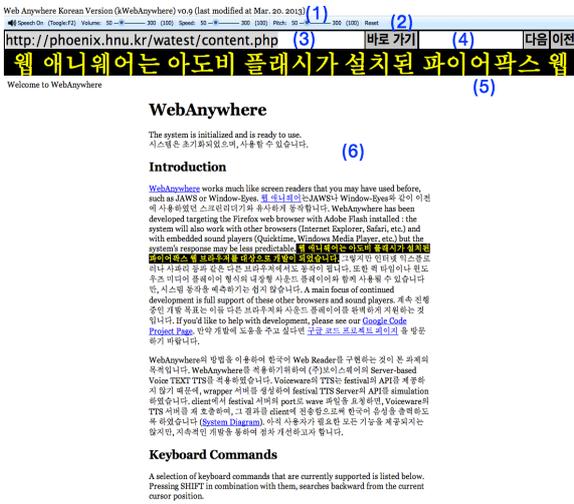
사용자 인터페이스를 적용한 kWebAnywhere의 실행화면은 <그림 4>에 표시하였다. 실행 화면의 최상단에 (1) 버전 정보를 표시하였으며, (2) 도구 바, (3) 직접 URL 이동, (4) 검색

5) Escape/Unescape는 HTML문서나 URL 등에서 HTML 명령어와 혼동되는 것을 피하기 위하여 특수 문자를 미리 정해진 기호나 열로 변환/역 변환하는 것을 말한다.

창, (5) 현재 출력되고 있는 텍스트 표시, (6) 웹 문서 내용 순으로 구성하였다. 검색 창이 검색하고자 하는 단어를 입력하면 단어를 포함하는 첫 번째 문장으로 이동하며, 다음 버튼과 이전 버튼으로 검색된 문장 간의 이동이 가능하다.⁶⁾

표 6. *kWebAnywhere*에 적용된 도구 바 버튼 및 기능
Table 6. Toolbar buttons applied in *kWebAnywhere* and their functions

도구 바 버튼	기능
Speech On (Toggle:F2)	음성 출력 on/off
Volume: 50 —●— 300 (100)	발성 크기 조절
Speed: 50 —●— 300 (100)	발성 속도 조절
Pitch: 50 —●— 300 (100)	발성 피치 조절
Reset	기본 값 복원



(a) 전체 구성도(Overall layout)



(b) 상단 영역 확대(Magnification of top area)

그림 4. *kWebAnywhere* 동작 화면
Figure 4. Screenshot of *kWebAnywhere*

6) TTS 시스템의 음성 출력 및 발성 속도 등을 제어하는 사용자 제어 인터페이스를 제외한 기능 및 디자인은 *WebAnywhere* 베타 버전을 적용하였다.

4.3 한계 및 연구 고려 사항

본 연구에서 구현한 *kWebAnywhere*는 *WebAnywhere*를 개선하였기 때문에 3.2절에서 기술한 한계를 모두 가지게 된다. 3.2절의 *WebAnywhere* 한계는 음성 출력 방식의 제한, 음성 인식기에 의한 입력 미지원, 프락시 방식에 의한 캐시 구현, 스크립트 언어 기반의 브라우저 기능 이용으로 발생하는 문제이다. 이것은 서버 기반의 웹 리더가 가지는 구조적인 한계일 수 있다.

*kWebAnywhere*는 웹 문서에 영어와 한글 등이 같이 존재하는 경우 문장에 따라 서로 다른 합성 엔진을 사용하고 있다. (예; 한국어 유미, 영어 Julie 엔진 사용) 따라서 한 페이지가 영어와 한글로 같이 작성되어 있는 경우, 사용자에게 따라 입력 문장에 따라 자동으로 전환되는 영어 음성 합성기와 한국어 음성 합성기의 음성이 부자연스러울 수 있다. 이런 점을 고려하여 본 연구자는 음성 합성음에 대하여 비슷한 음색을 표현할 수 있도록 연구를 지속하고 있으며[26], 이를 발전시켜 동일한 사람이 영어와 한국어를 같이 발성하는 것처럼 느낄 수 있도록 음성 변환 연구의 개선이 필요하다.

웹 문서를 음성으로 출력하기 위해서는 각 문장을 음성 합성기에 전송하여야 한다. 본 연구에서 채택한 *WebAnywhere*는 *WebProxy* 기능을 이용하여 이미 생성된 음성에 대하여 다시 음성 합성기를 호출하지 않고 이미 생성된 음성 파일을 재사용함으로써 음성 합성기가 설치된 서버의 부하와 네트워크 트래픽을 줄이도록 설계되었다. 만약 문장의 길이가 길어진다면 서로 다른 문장의 수가 늘어나기 때문에 캐시의 효과가 줄어들고, 문장의 길이가 작아진다면 캐시의 효과는 증가하나 음성 합성기에 전달되는 텍스트의 양이 줄어들어 음성 자연성이 저하될 수 있다. 따라서 적절한 문장의 길이는 캐시 능력과 네트워크 트래픽 양, 그리고 음성의 자연성 등에 영향을 줄 수 있다. 현재 구현된 *kWebAnywhere*에서는 영어의 경우 문장 부호와 접속사 등에 의하여 문장을 분리하고, 한글의 경우에는 문장 부호에 의해서만 문장을 분리한다. 영어와 달리 한글의 경우 복문이라 할지라도 문장 부호가 사용되지 않는 경우가 많아 문장 길이가 긴 경우가 많이 발생한다. 따라서 형태소 분석 등의 문장 분석 과정을 통하여 영어와 한글의 띄어 읽기 (문장 분리) 등의 연구가 고려되어야 한다.

5. 요약 및 결론

모바일 환경의 발달과 인터넷 기술의 발전으로 많은 사람들이 인터넷에 접속하여 정보를 검색하고 제공하고 있다. 그러나 기술의 빠른 발전 속도는 노약자나 장애인들에 대한 배려를 충분히 고려하고 있지 않아, 많은 사용자들이 첨단 기술의 이용에서 소외되고 있는 실정이다. 본 연구는 이들 중 저시력자나맹인 또는 컴퓨터 화면을 지속적으로 응시할 수

없는 사용자들을 대상으로 웹 정보의 원활한 접근을 목적으로 하고 있다.

최근 들어 정부나 많은 사회단체에서 장애인들을 대상으로 재정적 지원이나 IT 기기를 지원하고 있으나, 거의 대부분 개인 장비에 국한되고 있다. 모든 장애인들이 개인 장비를 휴대하고 외출하는 것이 아니기 때문에, 위급 상황이나 급하게 컴퓨터를 이용하여 인터넷 정보를 검색할 경우 프로그램의 설치나 다른 사람의 도움을 받아야 한다. 또한 개인이 이용하는 스크린 리더와 같은 프로그램의 경우에도 운영체제나 프로그램이 개선이 된 경우, 지속적인 유지 관리가 필요하다. 이런 점을 고려하여 본 연구는 클라이언트(컴퓨터)에 특정 프로그램의 설치 없이 유지보수를 편하게 하고, 비용을 적게 필요로 하는 서버기반의 웹 리더를 고려하였다.

서버기반의 웹 리더를 구현하기 위하여 2008년에 발표된 *WebAnywhere*를 기초로 하여 한국어 및 다국어 TTS를 이용한 *kWebAnywhere* 시스템을 구현하였다. *kWebAnywhere* 시스템은 *Festival* 음성합성기와 호환되도록 서버에 *Festival* 변환기를 구축하였으며, 다양한 스크립트 언어 간의 한국어 통신 문제를 해결하기 위하여 TTS 인터페이스의 설계 및 구현, 음성 합성음의 효과적인 제어를 위한 사용자 인터페이스를 설계 및 구현하였다.

본 연구에서 개발된 *kWebAnywhere*가 효과적으로 운영된다면 많은 저시력자나 장애인들이 외부의 공공장소의 컴퓨터 등에서 프로그램의 설치나 추가 비용 없이 인터넷에 쉽게 접속할 수 있을 것으로 보인다.

그러나 현재 개발된 *kWebAnywhere*는 입력되는 문장에 따라 다른 합성 엔진을 호출하기 때문에, 한국어나 영어의 경우 다른 사람이 텍스트를 읽어주는 느낌을 받아, 한국어와 영어의 음색을 통일하여 한 사람이 한국어와 영어를 읽어주는 음성 변환 연구도 진행되고 있다. 일반 음성 변환의 경우 음성 합성 시스템에서 음성 변환 작업이 이뤄지나 현재 진행되고 있는 음성 변환 연구의 경우 최종적인 음성 합성음에 대하여 적용하기 때문에 음질 조절 변수나 시간 등에 제약이 있어 음질 향상을 위한 추가 연구가 필요하다. 또한, 문장 단위로 음성 합성이 이뤄지기 때문에 선택된 문장의 길이에 따라 음성 합성음의 음질 및 네트워크, 시스템 부하 등이 예상되기 때문에 합성음의 음질을 고려한 한국어/영어의 문장 띄어 읽기에 대한 연구도 필요하다고 판단된다.

참고문헌

- [1] United Nations. (2006). International convention on the rights of persons with disabilities, Retrieved from <http://www.un.org/disabilities/documents/ppt/crpdbasics.ppt> on March 4, 2013.
- [2] Brault, M. W. (2012). Americans with disabilities: 2010, *United States Census Bureau*, 70-131.
- [3] Employment Development Institute. (2011). 2011 Survey on statistics of disabled, *Survey 2011-03*. (고용개발원 (2011). 2011 장애인 통계, 고용개발원 조사통계 2011-03.)
- [4] Wikipedia (2013). Accessibility, Retrieved from <http://en.wikipedia.org/wiki/Accessibility> on January 14, 2013.
- [5] Hyun, J. (2013). Web Accessibility & IT Transformation, Retrieved from <http://jhyun.wordpress.com> on May 6, 2013. (현준호 (2013). 웹 접근성 & IT를 통한 변화, <http://jhyun.wordpress.com>.)
- [6] Henry, S. L. et al. (2005). Introduction to Web accessibility, Retrieved from <http://www.w3.org/WAI/intro/accessibility.php> on May 1, 2013.
- [7] Henry, S. L. (2013). Mobile accessibility, Retrieved from <http://www.w3.org/WAI/mobile/> on March 18, 2013.
- [8] Berners-Lee, T. (1997). World Wide Web Consortium, Oct. 22
- [9] Henry, S. L. & McGee, L. (2013). Accessibility, Retrieved from <http://www.w3.org/standards/webdesign/accessibility> on March 25, 2013.
- [10] WebAIM. (2013). Introduction to Web accessibility, Retrieved from <http://webaim.org/intro/#types> on March 11, 2013.
- [11] Freedom Scientific. (2013). JAWS for Windows Screen Reading Software, Retrieved from <http://www.freedomscientific.com/products/fs/jaws-product-page.asp> on March 25, 2013.
- [12] XVision Technology. (2012). Sense Reader, Retrieved from <http://www.xvtech.com/xvtech> on March 25, 2013.
- [13] GW Micro. (2013). Window Eyes, Retrieved from <http://www.gwmicro.com/Window-Eyes/> on March 25, 2013.
- [14] Chen, C. L. (2008). FireVox: A screen reading extension for Firefox, Retrieved from <http://www.firevox.clcworld.net/> on March 25, 2013.
- [15] Google. (2013). ChromeVox, Retrieved from <http://www.chromevox.com/> on July 8, 2013.
- [16] Stony Brook University. (2012). HearSay 3: a vision for the blind, Retrieved from <http://www.cs.sunysb.edu/~hearsay/index.htm> on July 8, 2013.
- [17] Miyashita, H. & Sato, D. & Takagi, H. & Asakawa, C. (2007). aiBrowser for multimedia – Introducing multimedia content accessibility for visually impaired users, In Proc. of *ASSETS '07*, 91-98.
- [18] Bigam, J. P. (2008). WebAnywhere, Retrieved from

- <http://webanywhere.cs.washington.edu/> on September 10, 2012.
- [19] Voiceware Corp., VoiceText™, Retrieved from <http://www.voiceware.co.kr/kor/product/product1.php> on December 19, 2013.
- [20] Bigham, J. P. & Prince, C. M. & Ladner, R. E.(2008). WebAnywhere: A screen reader on-the-go, In Proc. of *the International Cross - Disciplinary Conference on Web Accessibility(W4A)*, pp. 73-82.
- [21] Schiller, S. (2007). Sound Manager 2, Retrieved from <http://www.schillmania.com/projects/soundmanager2/> on January 14, 2013.
- [22] Taylor, P. & Black, A. & Caley, R. (1998). The architecture of the festival speech synthesis system, In *Proc. of the 3rd ESCA Workshop in Speech Synthesis*, pp. 147-151, Jenolan Caves, Australia.
- [23] Arif, A. (2007). PHPProxy, Retrieved from <http://www.sourceforge.net/projects/poxy/> on January 14, 2013.
- [24] DHTMLX, JavaScript UI Library (2012). Ajax Components & HTML5 Framework, Retrieved from <http://dhtmlx.com/> on March 4, 2013.
- [25] Black, A. W. & Taylor, P & Caley, R. (1999). The Festival speech synthesis system, Retrieved from <http://www.cstr.ed.ac.uk/projects/festival/manual/> on July 10, 2013.
- [26] Yun, Y.-S. & Ladner, R. E. (2013). Bilingual voice conversion by weighted frequency warping based on formant space, In Proc. of *the 16th International Conference on Text, Speech and Dialogue (TSD2013)*, LNAI 8082, 137-144.

• **윤영선(Yun, Young-Sun)**

한남대학교 정보통신공학과
 대전시 대덕구 한남로 70 (오정동)
 Tel: 042-629-7569 Fax: 042-629-7843
 Email: ysyun@hannam.kr
 관심분야: 음성인식, 음성모델링, 음성처리, 웹 접근성,
 내장형 시스템 등
 2001 ~ 현재 한남대학교 교수