# HTMLtoVoiceXML 변환기의 설계 및 구현
## (Design and Implementation of a HTMLtoVoiceXML Converter)

최 훈 일 † 장 영 건 ††
(Hoon il Choi) (Young Gun Jang)

**요 약** 음성 기술의 발달과 VoiceXML 1.0의 등장으로 인하여 표준화된 방식으로 이동 단말기와 전화를 통해 음성으로 웹 컨텐츠에 접근할 수 있게 되었다. 거의 모든 웹 컨텐츠들은 HTML로 작성되어 있으며, 기존의 HTML로 작성된 수많은 웹 컨텐츠에 음성으로 접근하기 위해서는 HTML 문서들을 VoiceXML 문서로 변환하여야 한다. 이를 수동으로 변환하기 위해서는 많은 시간과 비용이 필요하게 된다.

본 논문에서는 이 문제를 해결하기 위하여 HTML 문서를 VoiceXML 문서로 자동 변환하는 HTMLtoVoiceXML 변환기를 설계하고 구현하였으며, 그 구조를 제시하고, 웹 페이지에서 중요한 내용을 포함한 조각을 지정하는 실용적인 알고리즘을 제안한다. 국내의 400 여 개의 웹 페이지를 대상으로 개발된 변환기의 성능을 시험하였고, HTML문법을 정확히 지키지 않은 경우를 제외하고, 거의 모두가 VoiceXML문서로 변환되어, 유효성과 실용성이 검증되었다.

***Abstract*** It is possible to access web contents by mobile terminals and telephones due to the evolution of voice and VoiceXML technologies. Nevertheless, because these days most of all the web contents are constructed by HTML, it is impossible to access them by using the new technology. Therefore, to access the contents through voices requires the conversion of the web document from HTML to VoiceXML, but its manual conversion process should be involved additional time and expenditure.

In this paper, we design and implement HTMLtoVoiceXML converter, suggest a system structure of it and propose a practical identification algorithm of meaningful fragmented group of contents to solve the problem. To test the performance and validity of the converter, we apply it to more than 400 web pages in Korean web sites, it works well except for web pages which involve invalid HTML codes.

## 1. 서 론

1990년 WWW(World Wide Web)이 소개된 이후 웹은 엄청난 속도로 확산되기 시작하여 오늘날 거의 모든 정보들은 웹을 통해 얻을 수가 있게 되었다. 이러한 웹 정보들은 HTML(HyperText Markup Language)이라는 마크업 언어를 통하여 웹 문서로 작성되어 넷스케이프나 익스플로러와 같은 웹 브라우저에 의해 해석되고, 컴퓨터 모니터를 통하여 전달된다.

오늘날, 무선 인터넷의 발달과 음성 기술의 발달로 웹 브라우저를 통하여 컴퓨터 모니터 상에서만 얻을 수 있었던 웹 정보를 이동 단말기와 전화를 통해서도 얻을 수 있게 되었다. 이동 단말기나 전화를 통하여 웹 정보를 얻기 위해서는 웹 정보가 HTML이 아닌 다른 마크업 언어로 작성되어 있어야 한다. 표준화된 방식을 사용하여 음성으로 웹 정보를 제공하기 위해서는 W3C에서 제안한 VoiceXML(Voice eXtensible Markup Language) [1]이라는 마크업 언어를 이용하여 문서를 작성하여야 한다. 하지만 HTML로 작성된 기존의 수많은 웹 정보를 VoiceXML 문서로 변경하여 제공하는 것은 많은 비용의 투입과 시간의 낭비를 초래한다. 이런 문제를 해결하기 위해서는 기존의 HTML을 그대로 유지하면서 필요시 HTML 문서를 자동으로 VoiceXML 문서로 변환하면 될 것이다. 이러한 변환기에 대한 연구는 국내는 물론이고, 외국에서도 발표된 것이 거의 없다.

† 학생회원 : 청주대학교 전산정보공학과
choihi@chongju.ac.kr
†† 정 회 원 : 청주대학교 컴퓨터정보공학과 교수
vgjang@chongju.ac.kr
논문접수 : 2001년 9월 4일
심사완료 : 2001년 11월 16일

IITML 문서를 VoiceXML 문서로 변환하기 위해서는 우선적으로 해결하여야 할 몇 가지 문제점들이 있다.

첫째, HTML 태그는 정보의 시각적 표시 방법만을 나타낼 뿐 XML 태그처럼 정보에 대한 의미를 포함하고 있지 않기 때문에 컨텐츠를 분리하기가 어렵다.

둘째, HTML과 VoiceXML의 정보 제공 방법의 차이로 인해 HTML 태그와 VoiceXML 태그는 서로 사용 용도가 확연히 다르다. 따라서 HTML 태그와 VoiceXML 태그는 1:1 매칭으로 변환할 수 없다.

셋째, 하나의 웹 페이지에서 시각적으로 제공되는 정보량과 음성적으로 제공되는 정보량에 차이가 발생한다. 즉, 청각정보는 시간에 따라 변하며, 그 정보를 기억해야 하기 때문에 단일 페이지에서 제공되는 정보의 양은 매우 제한적이다. 따라서, 하나의 HTML 문서를 하나의 VoiceXML 문서로만 변환 할 수 없다. 즉, 하나의 HTML 문서에 대해 여러 개의 VoiceXML 문서가 생성되어야 한다.

본 논문에서는 위에서 열거한 문제점들을 해결하기 위하여 변환 가능한 HTML 문서 형태를 게시판처럼 정보의 형태가 비슷한 내용들이 나열되어 있는 문서들로 제한하고, 이런 형태의 HTML문서를 해당 웹 페이지의 구성요소를 분석하여 변환 컨텐츠 후보로 분리하고, 후보 컨텐츠의 내용들을 분리하여 추출하고, 이를 Voice XML 문서의 형식으로 재구성하는 자동화 시스템인 HTML toVoiceXML 변환 시스템을 설계하고 구현하였다.

## 2. 관련 연구

### 2.1 VoiceXML

음성 인식/합성 기술이 발달함에 따라 웹에 접속하기 위한 사용자 인터페이스가 키보드 및 전화기의 키패드에서 음성으로의 전환이 거론되기 시작했다. 특히, AT &T, IBM, 루슨트 테크놀로지, 모토롤라 등 정보통신 분야의 4개 거대 기업이 모여 VoiceXML 포럼[2]을 설립하여, 1999년 8월 웹 컨텐츠와 웹 기반의 응용 등을 대화식 음성 응답을 통해 제공받을 수 있는 음성 마크업 언어인 VoiceXML 0.9 버전을 발표하였고, 2000년 3월 0.9 버전을 보완한 1.0 버전을 W3C에 제안하여 W3C에서 2000년 5월 VoiceXML 1.0을 웹의 대화형 마크업 언어 표준으로 발표하였다. 이로 인해 많은 업체들이 VoiceXML에 대해 관심을 갖기 시작하였고, 많은 연구가 활발히 진행되고 있는 상태이다. 외국의 경우 Bevocal, Nuance, Tellme, VoiceGenie 등에서 Voice XML 해석기를 개발하였으며, 몇몇 업체에서는 Voice XML 시범 서비스를 제공하고 있다. IBM에서는 Voice XML 응용을 개발할 수 있는 도구를 제공하고 있으나 표준을 완벽하게 구현하고 있지는 못하다. 국내에서는 성신여자대학교에서 VoiceXML 해석기의 구현에 관한 연구[3,4]를 수행하였으며, 미디어포드[5]와 브레인21[6, 7]에서도 VoiceXML 해석기를 개발하였다. 그러나 아직까지는 거의 모든 컨텐츠가 HTML로 작성되어 있어, 그 이용률이 매우 저조하다. VoiceXML이 이용할 수 있는 단말기의 대중성과 이동성, 대화형이라는 편의성 등의 많은 장점에도 불구하고, 대중화가 되지 않은 가장 큰 이유는 이용할 수 있는 컨텐츠와 서비스의 부족, VoiceXML로 웹 저작을 할 수 있는 전문인력의 부족 등이 있다. 따라서 기존의 IITML로 작성된 웹 컨텐츠를 자동으로 VoiceXML로 변환하여 서비스할 수 있는 기술이 요구된다.

### 2.2 HTML 코드 변환

VoiceXML 1.0의 등장이 2000년 5월이므로, HTML을 VoiceXML로 변환하는 연구는 발표된 것이 거의 없다. VoiceXML의 등장 전에 그 원형 중에 하나인 VoxML이 모토로라에 의하여 발표되었으며, HTML을 VoxML로 변환하려는 연구가 Goose 등에 의하여 최초로 이루어졌다. Goose 등은 WWW의 전형적인 3층 구조에 HTML을 VoxML로 변환하는 기능을 갖는 VoxML-Agent를 추가하여 클라이언트, 에이전트, 웹 서버, 데이타베이스의 4층 구조를 갖는, 전화기를 사용하여 웹 접근이 가능한 Vox 포탈에 대한 연구를 발표하였다. VoxML-에이전트에 대해서는 Vox 포탈과의 상호 작용에 대하여 중점적으로 언급되어 있고, 핵심 부분인 변환 기능의 설계에 대한 내용은 없다[8].

시각적 인터페이스를 주 수단으로 하는 HTML문서를 해석하여 음성 인터페이스를 부여하는 연구는 최근에 이루어지고 있으며, 보편적 접근을 위한 인터넷 내용의 적응 및 접근성 향상 도구를 위주로 한 문법적 속성에 근거한 코드변환[9,10], HTML의 구조적 성질에 기반한 코드 변환[11-14], 의미론적 코드 변환[15], 수작업에 의한 주석코드 추가[16,17] 등의 접근이 이루어지고 있으나, 좀더 쉬운 항해와 이해를 위하여 웹 페이지의 재배치, 주석 달기 및 웹 객체를 변환하는 것을 목적으로 하고, 특정한 사용자 그룹에 대한 특별한 필요성의 해결을 목표로 하고 있어 총체적 일반화에는 이르지 못하고 있다. Asakawa는 전화기 인터페이스를 갖는 주석 기반 프록시 서버에 관한 연구에서 HTML문서를 해석하여 문서를 완전히 변환하는 기술은 존재하지 않으며, 문서 저작자의 의도를 언어로 표현하기 위해서는 부가적인 주석을 사용할 수밖에 없다는 것을 강조하였다[16].

HTML코드만을 대상으로 시각적으로 표현되는 레이아웃을 추정하고, 각각의 내용들을 분리해 내는 것은 컴퓨터는 물론이고, HTML에 대한 지식을 가진 인간도 완벽하게 처리할 수 없다. 따라서 변환기를 구성하는 핵심 기술은 HTML문서의 통계적, 문법적, 구조적 특성을 활용하여 내용을 분리하는 것이다. 이에 대한 연구로는 Embley[18]와 Buttler[19]의 연구가 있다. Embley와 Buttler는 멀티 컨텐츠를 갖는 HTML 문서에서 게시판이나 검색결과처럼 비슷한 형태의 내용이 나열되어 있는 멀티 컨텐츠를 내용 단위로 분리하는 방법을 제시하였으며, 그 방법으로써 HTML 문서의 구조적 특징을 이용하여 내용 단위 분리의 기준이 되는 분리 태그를 추출하는 휴리스틱 알고리즘들을 제안하였다. 본 논문에서 구현한 HTMLtoVoiceXML 변환기는 Buttler가 제안한 휴리스틱 알고리즘을 참고하여 분리 태그를 추출하였다. 그러나 분리 태그를 기준으로 컨텐츠를 분리하는 과정과 분리된 컨텐츠에서 적절한 컨텐츠를 추출하는 과정은 Embley와 Buttler는 구체적인 언급이 없다.

## 3. HTMLtoVoiceXML 변환기 설계 및 구현

### 3.1 변환 가능한 HTML 문서 유형

오늘날 웹 저작자는 다양한 시각적 효과를 사용하여 가능한 많은 정보를 한 페이지에 표현하고자 한다. 이 정보는 시각적으로 단체화하여 조각나 있다. 즉, 메뉴, 광고, 내용 등의 조각이 하나의 페이지에 표현된다. 예를 들면 그림 1과 같은 웹 페이지는 A, B, C, D, E, F, G의 조각이 하나의 페이지에 표현된다. 이 조각 중에서 어느 조각이 이 페이지에서 제공하고자 하는 주요 정보인지를 판단하고, 해당 정보를 내용 단위로 분리하여, 그 내용 단위를 기준으로 VoiceXML 문서로 변환하여야 한다.

VoiceXML은 음성적 정보 제공 방법을 이용하기 때문에 HTML 페이지처럼 한 페이지에 많은 정보를 표현할 수 없다. 그래서 HTML을 VoiceXML로 변환하기 위해서는 먼저, HTML 페이지를 조각 단위로 나누고 이 조각 단위로 VoiceXML 문서를 생성해야 한다. 이러한 문제는 HTML을 VoiceXML로의 변환에만 문제되는 것이 아니라, HTML을 무선 인터넷용 마크업 언어로 변환 할 때에도 발생되는 문제점이다. 그러나 앞서 언급했던 것처럼 HTML 태그는 정보의 시각적 표시 방법만을 나타낼 뿐 XML 태그처럼 정보에 대한 의미를 포함하고 있지 않기 때문에 조각 단위로 분리하기가 어렵다.

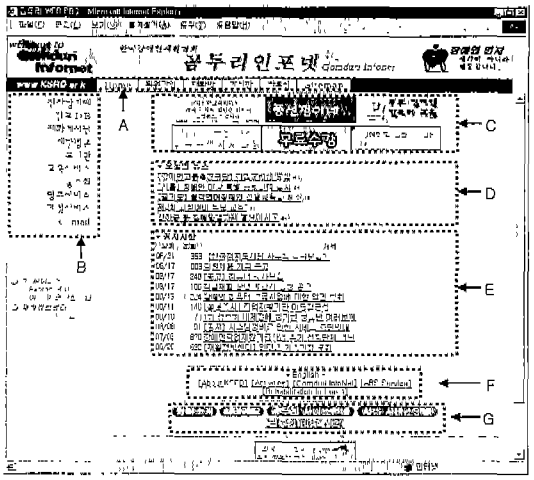


그림 1 웹 페이지 형태

따라서 본 논문에서는 HTML 문서의 구조 분석을 통해 컨텐츠의 조각을 분리할 수 있는 HTML 문서의 유형을 제안하고, 이런 문서 유형을 변환 대상으로 한다. 본 논문에서 제안한 변환 가능한 HTML 문서 유형은 비슷한 컨텐츠가 나열되어 있는 형태로써 그림 2와 같이 게시판 유형, 리스트 유형, 검색결과 유형으로 나눌 수 있다. 이런 유형의 HTML 문서는 트리 구조로 표현할 때 동일한 형태의 자식 노드를 많이 갖고 있고, 그 내용 속에 많은 문자를 포함하고 있는 점을 이용하여 컨텐츠의 위치를 구할 수 있다. 그림 1에서 Voice XML로 변환 가능한 내용은 공지사항(E)과 오늘의 뉴스(D)이다.

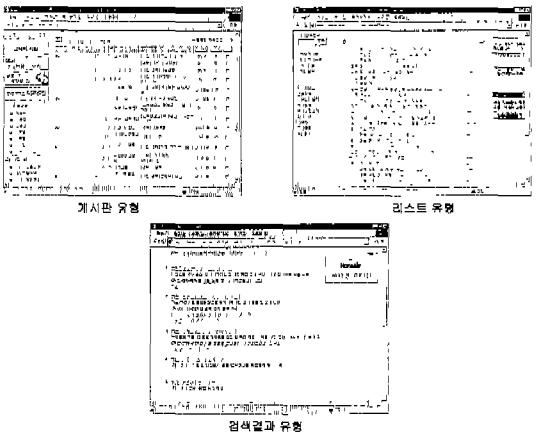


그림 2 변환 가능한 HTML 문서 유형

### 3.2 변환기 시스템 구조

본 논문에서 구현한 HTMLtoVoiceXML 변환기는 해당 URL의 웹 페이지를 읽어 웹 페이지에서 HTML 페이지의 구조 분석을 통해 적절한 컨텐츠를 추출하여 이 추출된 컨텐츠를 대상으로 하여 음성 시나리오를 생성하고 이를 VoiceXML 문서로 변환한다. HTMLto VoiceXML 변환기는 크게 HTML 구조 분석을 통한 컨텐츠 추출과정과 VoiceXML 문서 생성과정으로 나눌 수 있으며 전체 시스템 구조는 그림 3과 같다.

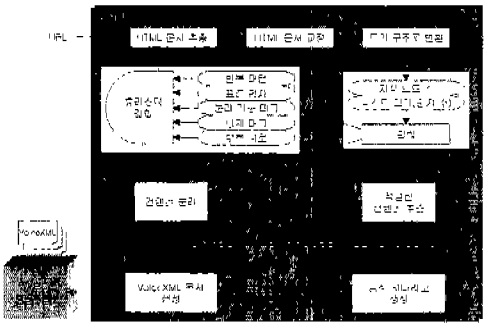

그림 3 HTMLtoVoiceXML 변환기 구성도

### 3.3 HTML 구조 분석을 통한 컨텐츠 추출

컨텐츠를 추출하는 방법은 게시판과 같이 정보의 형태가 비슷한 컨텐츠가 나열되어 있는 HTML 문서들을 HTML 저작툴을 이용하여 작성하거나 또는 직접 문서를 작성할 때 나타나는 몇 가지의 일반적인 규칙들을 이용하여 컨텐츠를 분리하기 위해 경계가 되는 태그를 추출하여 추출된 태그를 기준으로 컨텐츠들을 분리하고 분리된 컨텐츠들 중에서 적절한 컨텐츠를 추출해낸다. 컨텐츠를 추출하는 전체 흐름도는 그림 4와 같다.

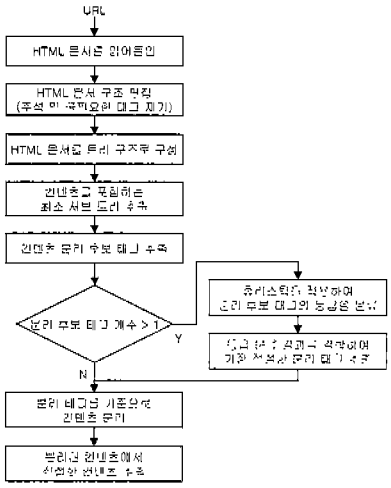


그림 4 컨텐츠 추출 흐름도

3.3.1 HTML 문서 트리 구조로 구성

읽어들인 HTML 문서는 트리 구조로 구성한다. HTML 문서를 트리 구조로 구성하는 이유는 트리 구조가 HTML 문서 구조를 분석하기가 좀더 용이하기 때문이다. HTML 문서를 트리 구조로 구성할 때 DOM (Document Object Model)을 사용하면 쉽게 트리화 할 수 있다.

3.3.2 컨텐츠를 포함하는 최소 서브 트리 추출

컨텐츠를 포함하는 최소 서브 트리란 전체 트리 구조에서 컨텐츠를 포함하며 크기가 가장 작은 트리를 말한다. 컨텐츠를 추출하기 전에 컨텐츠가 포함되어 있는 최소 서브 트리를 먼저 추출하는데 이 최소 서브 트리를 추출하기 위해 각 노드의 자식 노드 수를 구하여 가장 많은 자식 노드를 갖는 노드를 루트로 하는 트리를 최소 서브 트리로 한다. 왜냐하면 컨텐츠가 나열되어 있는 형태의 웹 페이지에서는 자식 노드의 수가 가장 많은 노드를 루트로 하는 서브 트리에 컨텐츠가 포함되어 있을 확률이 가장 많기 때문이다.

그러나 최소 서브 트리를 추출할 때 자식 노드의 개수만을 고려한 경우 메뉴가 많은 웹 페이지에서는 가장 많은 자식 노드를 갖는 노드를 루트로 하는 최소 서브 트리는 메뉴를 포함하는 서브 트리가 되기 때문에 최소 서브 트리를 구할 때 자식 노드의 개수만을 고려하는 것은 그리 좋은 방법이 아니다.

본 논문에서는 많은 자식 노드를 갖는 노드의 내부 텍스트 크기를 이용하여 최소 서브 트리를 추출하는 방법을 제안한다. 즉, 자식 노드를 많이 가지고 있는 노드들을 구하여 이 노드 내에 있는 텍스트의 크기(문자열의 수)를 구하여 텍스트의 크기가 가장 큰 노드를 루트로 하는 서브 트리를 최소 서브 트리로 한다. 이 방법은 메뉴를 포함하는 서브 트리는 자식 노드의 개수는 많지만 텍스트의 크기는 크지 않다는 점을 이용한 것으로써, 이 방법을 이용하면 메뉴와 같이 나타내는 텍스트의 크기가 작고, 자식 노드가 많은 서브 트리를 최소 서브 트리로 지정하는 오류를 피할 수 있다.

본 논문에서 제안하는 최소 서브 트리 추출 알고리즘은 다음과 같다.

1. 웹 페이지 트리 구조에서 최대 자식 노드 수 M을 구한다.
2. **repeat**

**if** 현재 노드의 자식 노드수 >= M * 0.5 **then**
현재 노드를 C 배열에 저장
**end if**
**until** end-of-file

3. C 배열에 저장되어 있는 각 노드를 루트로 하는 서브 트리를 구하여 이 서브 트리내의 텍스트의 크기를 S 배열에 저장
4. 최소 서브 트리 := S 배열에 저장된 값 중 크기가 가장 큰 값과 일치되는 노드를 루트로 하는 서브 트리
**/* S 배열에서 가장 큰 값이 2개 이상일 경우 */**
**if** Count(Max(S 배열의 값)) >= 2 **then**
최소 서브 트리 := 크기가 가장 큰 값 중 자식 노드 수가 가장 많은 노드를 루트로 하는 서브 트리
**end if**

3.3.3 컨텐츠 분리 후보 태그 추출

컨텐츠를 내용 단위로 분리하기 위한 분리 태그를 추출하기 위해 먼저 컨텐츠 분리 후보 태그들을 추출해야 한다. 분리 후보 태그는 최소 서브 트리의 루트 노드의 자식(Child) 노드를 컨텐츠 분리 후보 태그로 한다.

3.3.4 휴리스틱 알고리즘

컨텐츠 분리 후보 태그가 하나일 경우에는 그 후보 태그가 바로 컨텐츠를 분리하기 위해 경계가 되는 분리 태그가 되지만 둘 이상일 경우에는 분리 후보 태그들 중 적절한 태그를 분리 태그로 선택을 해야 한다. 여러 개의 분리 후보 태그 중에서 적절한 분리 태그를 추출하는 방법은 HTML 문서를 작성할 때 나타나는 몇 가지 규칙을 기반으로 하는 휴리스틱 알고리즘을 이용하여 분리 태그를 추출한다. 본 논문에서 사용한 휴리스틱(Heuristic)은 다음과 같다.

▶ 표준 편차 휴리스틱

표준 편차 휴리스틱은 각 분리 후보 태그들 사이의 글자 수에 대한 표준 편차를 구하여 계산 결과를 오름차순으로 나열하여 등급을 분류한다. 이 휴리스틱은 웹 문서에서 동일한 형태의 컨텐츠는 컨텐츠의 크기(글자 수)가 비슷하다는 점을 이용한 것이다.

▶ 반복 패턴 휴리스틱

반복 패턴 휴리스틱은 멀티 컨텐츠를 갖는 웹 문서에서 각 컨텐츠 사이에는 일관성 있게 반복적으로 나타나는 태그 쌍들이 존재할 수 있는데, 반복 패턴 휴리스틱은 이 태그 쌍을 이용하여 컨텐츠를 분리한다.

반복 패턴 휴리스틱 과정은 다음과 같다.

- 분리 후보 태그들 사이에 일반 텍스트가 없는 태그 쌍들의 발생횟수를 구한다.
- 태그 쌍의 발생횟수가 문서상의 각 분리 후보 태그의 발생횟수 중 가장 작은 것보다 10% 이상일 경우 다음 단계를 수행한다.
- 태그 쌍을 이루는 각 태그의 발생횟수를 구하여 태그 쌍의 발생횟수와의 차이에 대한 절대값을 구한다.
- 절대값의 오름차순으로 태그의 등급을 분류한다.
- 위의 방법을 반복하다보면 특정 태그는 등급 리스트에 여러 번 나올 수 있는데 이때 이 태그의 등급은 가장 좋은 것만을 채택한다.

▶ 부분 경로 휴리스틱

부분 경로 휴리스틱은 후보 태그 노드에서 임의의 다른 노드까지의 모든 경로를 목록화 하고 각각의 식별된 경로의 발생횟수를 계산한다. 이 때 임의의 다른 노드는 이 분리 태그 노드로부터 도달할 수 있고 선택된 서브 트리 내에 존재하는 노드이어야 한다.

후보 태그들은 모든 식별된 경로의 발생횟수의 내림차순으로 등급을 분류한다.

▶ 형제 태그 휴리스틱

형제 태그 휴리스틱은 최소 서브 트리에서 인접한 형제 태그 쌍의 발생횟수를 세어 내림차순으로 모든 태그 쌍들의 등급을 분류한다. 태그 쌍들의 발생횟수가 같은 경우, 태그 쌍의 등급은 웹 문서에서의 발생 순서를 따른다. 즉 웹 문서에서 앞에 있는 태그 쌍이 높은 등급을 받게 된다.

이 휴리스틱은 최소 서브 트리 내에서 분리 태그의 발생 횟수는 컨텐츠의 수와 동일하게 나타날 가능성이 많다는 점을 이용한 것이다.

▶ 분리가능 태그 휴리스틱

분리가능 태그란 웹 문서에서 컨텐츠를 분리할 때 경험적으로 자주 사용되는 태그를 말한다. 아래에는 이런 분리가능 태그들이 자주 사용되는 순으로 나열하여 놓았는데, 이런 태그들은 여러 웹 페이지를 통해 경험적으로 얻은 결과로 분류하였다.

분리가능 태그 휴리스틱이란 분리 후보 태그들을 분리가능 태그 표를 참조하여 등급이 높은 순서대로 재배열하여 분리 후보 태그들의 등급을 분류한다.

분리가능 태그 = {tr, table, p, li, hr, dt, ul, pre, font, dl, div, dd, blockquote, b, a, span, td, br, h4, h3, h2, h1, strong, em, i}

3.3.5 분리 태그 추출

위의 각 휴리스틱들은 특정 형태의 웹 문서에서 최적화되어 있고 또한 다른 휴리스틱들과 독립적이다. 그러므로 웹 문서의 올바른 컨텐츠 분리 태그를 추출할 가

능성을 향상시키기 위해 각각의 독립적인 휴리스틱을 결합시키는 것을 고려한다.

다섯 가지의 휴리스틱에 대한 최상의 결합 상태를 결정하기 위해, 스탠포드 확신도 이론(Stanford certainty theory)[20]을 이용한다. 이 이론을 사용하기 위해서는 각각의 개별적인 휴리스틱에 대한 확신도(certainty factor)를 가지고 있어야 한다. 각 휴리스틱에 대한 확신도는 표 1과 같으며, 이는 Buttler가 많은 웹 페이지를 대상으로 실험하여 얻은 결과를 참조하였다.

표 1 각 휴리스틱에 대한 확신도

| 휴리스틱/등급 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 표준 편차 | 0.78 | 0.18 | 0.10 | 0.00 |
| 반복 패턴 | 0.73 | 0.13 | 0.00 | 0.00 |
| 분리가능 태그 | 0.40 | 0.46 | 0.13 | 0.07 |
| 부분 경로 | 0.85 | 0.06 | 0.02 | 0.00 |
| 형제 태그 | 0.63 | 0.17 | 0.12 | 0.06 |

5개의 각 휴리스틱을 결합하는 방법에는 총 26가지가 있으나 5개의 휴리스틱을 모두 결합하여 분리 태그를 추출하는 것이 가장 정확한 방법이다.

▶ 스탠포드 확신도 이론

스탠포드 확신도 이론(Stanford certainty theory)은 신뢰도를(confidence measure) 정의하고 각각의 독립적인 증거를 결합하기 위한 간단한 규칙을 생성한다. 만약 두 개의 독립적인 관측의 증거가 같은 결과를 지지한다면 스탠포드 확신도 이론은 두 개의 독립적인 관측의 증거를 결합하기 위해 다음과 같은 규칙을 제공한다.

CF(E1)는 관측 B에 대한 증거 E1의 확신도 요소(certainty factor)라 하고, CF(E2)는 같은 관측 B의 증거 E2에 대한 확신도 요소라고 하면, B에 대한 새로운 확신도 요소 CF는 B의 복합 확신도 요소(compound certainty factor) 라고 하며, 다음 식에 의해 계산된다.

CF(E1) + CF(E2) - (CF(E1) X CF(E2))

3.3.6 분리 태그를 기준으로 컨텐츠 분리

분리 태그를 기준으로 컨텐츠를 내용 단위로 분리할 때 각 컨텐츠의 시작점과 끝점을 추출해야 한다. 즉, 분리 태그의 시작 태그와 바로 다음에 나타나는 분리 태그의 시작 태그를 컨텐츠의 시작점과 끝점으로 할 것인지, 아니면 분리 태그 이전의 내용부터 바로 다음에 나타나는 분리 태그의 시작 태그를 시작점과 끝점으로 할 것인지를 결정하여 각 컨텐츠의 내용을 분리해야 한다.

본 논문에서 제안하는 분리 태그를 기준으로 컨텐츠를 분리하는 과정은 다음과 같다.

1. 최소 서브 트리의 루트 노드의 첫 번째 자식 노드 N을 구한다.
2. **repeat**
   **if** (N == STYLE 태그 **or** SCRIPT 태그 **or** !(주석) 태그)
   **or** ( N == BR 태그 **and** BR 태그 ! = 분리 태그)
   **or** ( N == HR 태그 **and** HR 태그 ! = 분리 태그)
   **or** ( N == P 태그 **and** P 태그 ! = 분리 태그)
   **then**
   N := 다음 형제 노드
   **continue /* if문 다시 수행 */**
   **else**
   repeat 문 종료
   **end if**
   **until** end-of-최소 서브 트리
3. **if** N == 분리 태그 **then**
   컨텐츠 := 첫 번째 분리 태그의 시작 태그와 바로 다음에 나타나는 분리 태그의 시작 태그 사이의 내용
   **else**
   컨텐츠 := 첫 번째 분리 태그 이전의 내용부터 다음에 나타나는 분리 태그의 시작 태그 사이의 내용
   **end if**

3.3.7 분리된 컨텐츠 중에서 적절한 컨텐츠 추출

분리 태그를 기준으로 컨텐츠의 내용을 분리하면 적절하지 못한 내용들도 포함될 수 있다. 그래서 분리된 컨텐츠에서 적절한 컨텐츠만을 추출하여야 한다. 적절한 컨텐츠를 추출하는 방법은 컨텐츠가 나열되어 있는 웹 페이지의 각 컨텐츠의 텍스트는 숫자나 날짜, 문자들이 일정한 형태로 나타나게 되는데 이런 특징을 이용하여 적절한 컨텐츠를 추출한다.

본 논문에서 제안하는 적절한 컨텐츠 추출 방법은 다음과 같다.

1. repeat /* 모든 **분리된 컨텐츠를 대상으로 함** */
   분리된 컨텐츠의 텍스트 항목의 형태를 숫자형, 날짜형, 문자형으로 구분하여 각 컨텐츠의 텍스트 항목 형태 정보를 F 배열에 저장
   **until** end-of-분리된 컨텐츠
2. F 배열에 저장된 형태 정보 중 같은 형태의 개수

를 구함.

3. 개수가 가장 많은 형태 정보를 구하여 이 형태 정보를 갖는 컨텐츠를 추출

### 3.4 VoiceXML 문서 생성

3.4.1 음성 시나리오 생성

컨텐츠 추출과정을 통해 추출된 컨텐츠는 음성 인터페이스를 통해 사용자에게 정보를 제공하게 된다. 음성적 정보 제공 방법은 시각적 정보 제공 방법보다 사용자가 한번에 이해할 수 있는 정보의 양이 제한적이기 때문에 추출된 컨텐츠의 양에 따라 N 개의 음성 시나리오가 생성된다.

본 논문에서는 추출된 컨텐츠의 평균 문자 수가 100개 미만이면 4개의 컨텐츠로 하나의 음성 시나리오를 구성하고, 100개 이상이면 2개의 컨텐츠로 하나의 음성 시나리오를 구성하도록 하여 N개의 시나리오를 생성하도록 하였다.

생성되는 음성 시나리오의 기본 구조는 다음과 같다.

**컴퓨터** : 컨텐츠에 대한 음성 제공

**컴퓨터** : 선택 가능한 메뉴에 대한 안내 음성 제공

**사용자** : 메뉴 선택

**컴퓨터** : 해당 VoiceXML 문서로 이동

3.4.2 VoiceXML 문서 생성

각 음성 시나리오는 VoiceXML 1.0 형식에 맞춰 VoiceXML 문서를 생성한다. 생성되는 VoiceXML 문서의 기본 구조는 그림 5와 같다.

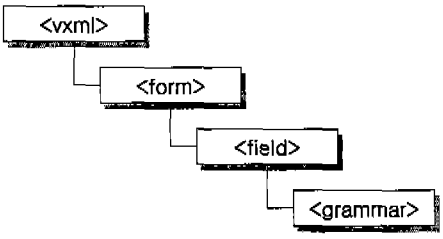


그림 5 생성되는 VoiceXML 문서의 기본 구조

## 4. 구 현

### 4.1 구현환경

본 논문에서 구현한 HTMLtoVoiceXML 변환기는 Windows NT4.0 환경에서 Platform SDK를 이용하여 VC++6.0으로 구현하였다. 그리고, HTML 문서를 파싱하기 위해 IE5.0이상에서 제공하는 MSHTML 라이브러리를 이용하였다.

### 4.2 변환 예

그림 6의 중앙일보 전체기사 페이지에는 40개의 뉴스가 리스트 형태로 나열되어 있는데, 이 페이지를 대상으로 논문에서 구현한 HTMLtoVoiceXML 변환기를 이용하여 변환한 결과는 다음과 같다.

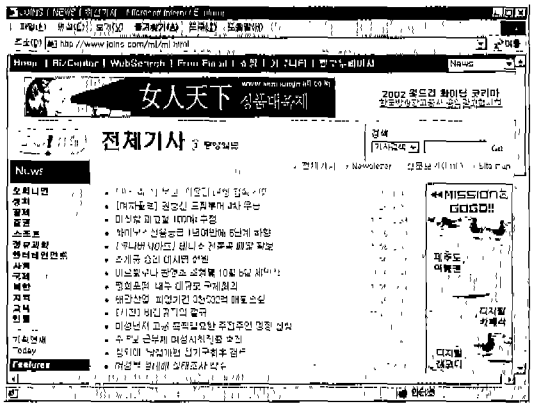


그림 6 중앙일보 전체기사 페이지

-추출된 컨텐츠 개수 : 40개

-생성된 VoiceXML 문서 개수 : 10개(파일명 : 0.vxml ~ 9.vxml)

다음은 생성된 10개의 VoiceXML 문서 중 첫 번째 VoiceXML 문서의 내용이다.

VoiceXML 문서)

```
<?xml version="1.0"?>
<vxml version="1.0">
<form id="page0">
<field name="field0">
<prompt>
1번  금감위 "서울보증보험 공적자금 2조1천억원
     추가투입"  10/15 11:33
2번  [일본의 최근 과거사 언급 내용]  10/15 11:31
3번  [전국체전] '비운의 유도스타' 윤동식
     선수생활 마감  10/15 11:30
4번  제1회 '해킹·바이러스 예방의 날' 실시  10/15
     11:29
</prompt>
<prompt>
상세정보를 원하시면 원하는 번호를 말씀해 주시고
다음 내용을 원하시면 다음을 말씀해 주세요.
</prompt>
<grammar>
```

```
1번 | 2번 | 3번 | 4번 | 다음
</grammar>
<filled>
<if cond="field0 == '1번'">
<goto next="http://www.joins.com/news/2001/10/
 15/all/20011015113334104230.html">
<elseif cond="field0 == '2번'">
<goto next="http://www.joins.com/news/2001/10/
 15/all/20011015113135104162.html">
<elseif cond="field0 == '3번'">
<goto next="http://www.joins.com/news/2001/10/
 15/all/20011015113023104400.html">
<elseif cond="field0 == '4번'">
<goto next="http://www.joins.com/news/2001/10/
 15/all/20011015112906104260.html">
<elseif cond="field0 == '다음'">
<goto next="1.vxml">
</if>
</filled>
</field>
</form>
</vxml>
```

## 5. 시험 및 평가

### 5.1 시험 대상 및 방법

본 논문에서 구현한 HTMLtoVoiceXML 변환기의 성능을 시험하기 위하여 일반적인 접근 빈도가 높고 본 논문에서 변환 대상으로 규정한 게시판 유형, 리스트 유형과 검색결과 유형을 갖는 구인/구직 사이트, 신문 사이트, 검색 사이트를 시험 대상 사이트로 결정하였다. 각각의 유형에 대하여 시험 대상으로 선정한 구체적인 사이트는 표 2와 같고, 제시한 3가지 유형을 포함하여 통합적으로 시험한 사이트는 표3과 같다.

시험 방법은 GIT[19]에서 적용한 방식과 동일하며, 적용한 알고리즘의 성공률을 두 단계로 계산한다. 첫째, 각 웹사이트에 대하여 적용된 페이지 중에서 제시된 알고리즘을 적용하여 최고 점수를 받은 태그가 정확한 분리 태그인 페이지의 비율을 계산한다. 둘째, 이 각각의 발견적 알고리즘과 통합된 알고리즘에 대한 성공률을 결정하기 위하여 전체 웹사이트에 대하여 평균을 구한다.

### 5.2 시험 결과

표 2에 열거된 사이트에 대한 분리 성공률은 약 200개의 웹 페이지에 대하여 적용한 결과, HTML을 문법에 맞지 않게 서술한 페이지를 제외하면 100%의 성공률을 보였다. 100%의 성공률을 보인 이유는 표 2에 열거된 웹사이트는 표로 구성되어 있는 경우가 많고, 하나의 분리 후보 태그만을 갖는 경우가 많아 휴리스틱 알고리즘을 적용할 필요가 없는 페이지가 많았기 때문이다.

표 2 시험 대상 웹사이트

| 구분 | 웹사이트 명 | 웹사이트 URL |
|---|---|---|
| 구인/구직 (게시판 유형) | 인크루트 | www.incruit.com |
| | 리크루토 | www.recruit.co.kr |
| | 헬로잡 | www.hellojob.com |
| | 잡엑스 | www.jobex.co.kr |
| | 데브피아 | www.devpia.com |
| 신문 (리스트 유형) | 동아일보 | www.donga.com |
| | 중앙일보 | www.joins.com |
| | 스포츠투데이 | www.sportstoday.co.kr |
| | 국민일보 | www.kukminilbo.co.kr |
| | 한겨레 신문 | www.hani.co.kr |
| 검색 (검색결과 유형) | MSDN | msdn.microsoft.com |
| | 한미르 | www.hanmir.com |
| | 알타비스타 | www.altavista.co.kr |
| | 구글 | www.google.com |
| | 네이버 | www.naver.com |

표 3 성공률 시험 대상 사이트

| 웹사이트 명 | 웹사이트 URL |
|---|---|
| 한국아이 | www.hankooki.com |
| 한국일보 | www.hankooki.com/hankook.htm |
| 일간스포츠 | www.hankooki.com/dailysports.htm |
| 서울경제 | www.hankooki.com/sed.htm |
| Korea Times | www.hankooki.com/sed.htm |

따라서 좀 더 다양한 웹 페이지 저작 기법을 사용한 웹 페이지를 대상으로 분리 태그 추출 성공률을 시험하였다. 표 4는 표 3에서 제시한 웹사이트에서 약 200 개의 웹 페이지를 대상으로 각각의 휴리스틱 알고리즘에 대한 확률적 순위와 휴리스틱 결합 알고리즘의 성공률을 표시하였다. GIT의 결합 알고리즘 성공률이 94%인 것과 비교해서 높은 성공률을 보인 이유는 적용한 웹 페이지의 유형이 GIT에 비하여 적은 것으로 보인다. GIT의 경우는 웹 페이지에 대한 정보가 없어 정확한 비교는 불가능하다. 시험 결과는 규칙성을 갖는 표나 탐

색 결과의 링크 나열과 같은 컨텐츠에 대해서는 구현된 변환기가 매우 안정적이라는 것을 보여주며, 좀 더 다양한 방식으로 표현한 웹 페이지의 경우도 통합 휴리스틱 알고리즘을 적용하였을 때 99%의 성공률을 보여 거의 모든 웹 페이지가 변환이 되는 것을 확인할 수 있다.

표 4 시험 데이타에 대한 각 휴리스틱 알고리즘의 확률적 순위와 결합 알고리즘의 성공률

| 휴리스틱 알고리즘 | 순위 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 분리가능태그 | 0.85 | 0.1 | 0.15 | 0.0 |
| 반복패턴 | 0.65 | 0.0 | 0.15 | 0.0 |
| 부분경로 | 0.95 | 0.0 | 0.0 | 0.0 |
| 형제태그 | 0.80 | 0.2 | 0.0 | 0.0 |
| 표준편차 | 0.70 | 0.2 | 0.1 | 0.0 |
| 결합알고리즘 | 0.99 | 0.02 | 0.01 | 0.0 |

다만, 한겨레신문의 전체기사보기 웹 페이지의 경우 원하는 컨텐츠 추출이 제대로 이루어지지 않았는데, 그 이유는 웹 페이지의 HTML 문서에서 BODY 태그 아래에 있는 A 태그의 시작 태그만 있고 종료 태그가 없어 HTML 문서를 DOM를 이용하여 트리 구조로 변환시 A 태그의 중첩 범위가 정확하게 해석되지 않았기 때문이다. 따라서 웹 컨텐츠를 분리하기 위해서는 저작자가 HTML 문법을 정확하게 준수할 필요가 있다.

아래 HTML 코드는 한겨레신문의 문제가 되는 코드 부분이다.

```
        :
        :
</head>

<body bgcolor=white text=black topmargin=0
 leftmargin=0 marginwidth=0 marginheight=0>
<a name="top"> <-- A 태그의 시작 태그에 대
 응하는 종료 태그가 없다.
<!--① family begin--->
<!--① family begin:none---->
<style type="text/css">
        :
        :
```

## 6. 결론 및 향후 과제

본 논문에서는 HTML로 작성된 기존의 컨텐츠를 이동단말기 및 전화를 통하여 음성으로 제공하기 위하여 VoiceXML 문서로 변환하는데 드는 비용과 시간을 줄이는 HTML 문서를 자동으로 VoiceXML 문서로 변환하는 HTMLtoVoiceXML 변환기를 설계하고 구현하였다.

개발된 변환기를 사용하여 국내의 약 200여 개의 웹 페이지를 대상으로 변환기의 기능과 성능을 시험하였다. 적용 결과 HTML 문법을 정확하게 적용한 모든 웹 페이지를 정확하게 VoiceXML 문서로 변환할 수 있었다. 따라서 기능의 유효성, 실용성 및 신뢰성이 있다고 판단된다.

몇몇 HTML 문법을 준수하지 않은 웹 페이지, 특히 시작 태그에 대한 종료 태그가 없는 경우는 제대로 변환되지가 않는데, 이를 해결하기 위해 HTML 문서를 XHTML 문서로 변환하여 이를 VoiceXML 문서로 변환하면 된다. XHTML로 변환할 수 있는 Tidy 유틸리티[21]는 한글의 경우 제대로 변환이 되지 않아 한글처리가 가능한 유틸리티가 개발될 필요가 있다.

또한, 현재는 하나의 웹 페이지를 대상으로 변환기를 설계하였지만, 웹 컨텐츠에 따라서는 동일한 성격을 가지면서 많은 웹 페이지로 구성되어 있는 컨텐츠가 많다. 따라서 좀 더 실용성을 가지려면 동일한 성격의 연속된 웹 페이지의 경우에는 한번의 접속만으로 해당 내용을 모두 변환하는 기능이 요구된다. 또한 HTML 문서들이 갖는 많은 문제점에 의해 변환 가능한 대상 HTML 문서의 형태를 제한하였는데, 변환 대상을 확장하기 위해서는 사전 지식을 이용한 방법 등 좀더 지능적인 방법과의 결합이 요구된다고 판단된다.

## 참 고 문 헌


[1] VoiceXML Forum, www.voicexml.org

[2] W3C, "Voice eXtensible Markup Language(Voice XML) version 1.0", http://www.w3.org/TR/voicexml, W3C Note 05 May 2000.

[3] 김경란, 홍기영, "VXML 편집기와 음성 브라우저의 설계 및 구현", 2000년 한국정보과학회 춘계학술대회 논문집, 27권 1호(B). pp414-416, 2000. 4.

[4] 김경란, "VoiceXML 기반 음성 브라우저의 설계 및 구현", 성신여자대학교 석사학위 논문, 2001. 2.

[5] http://www.mediaford.co.kr/

[6] 윤현주, 하준, 은성배 김병호, 강상민, 서원균, "VXML 인터프리터의 설계 및 구현", 제9회 한국음성과학회 학술발표대회 논문집, 2000.

[7] http://www.brain21.com/

[8] Stuart Goose, Mike Newman, Claus Schmidt, Laurent Hue, "Enhancing Web accessibility via the Vox Portal and a Web-hosted dynamic HTML<->VoxML converter", WWW9, Volume

33, Numbers 1-6, pp583-592, June 2000.
[9] Mohan, R., Smith, J. & Li, ,C.-S., Adapting multimedia internet content for universal access", IEEE Transactions on Multimedia, pp104-114, , 1999.
[10] Vorburger, M. Altifier, "Web acessibility enhancement tool", Available at http://www.vorburger.ch/, 1999.
[11] Asakawa,C. et al, "User Interface of a Homepage Reader", Pro. of ASSET'98, pp149-156, April 1998.
[12] Kaasinen, E. et al, "Two Approaches to bringing internet Services to WAP devices", Pro. of WWW9 Conference, pp231-246, May 2000.
[13] Raman,T.V., "Emacspeak- direct speech access", Pro. of ASSETS'96, pp32-36, April 1996.
[14] Zajicek M, "A Web navigation tool for the blind", Pro. of ASSETS'98, pp204-206, April 1998.
[15] Anita W. Huang, "A Semantic Transcoding System to Adapt Web Services for Users with Disabilities", Pro. of ASSETS'00, pp156-163, Nov. 2000.
[16] Asakawa, "Annotation-Based Transcoding for Nonvisual Web Access" Pro. of ASSETS'00, pp172-179, Nov. 2000.
[17] Hironobu Takagi, "Transcoding Proxy for Nonvisual Web Access", Pro. of ASSETS'00, pp164-171, Nov. 2000.
[18] D.W. Embley, Y.S. Jiang, and Y.-K. Ng. "Record-boundary discovery in Web documents", In Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (SIGMOD'99), pp467-478, Philadelphia, Pennsylvania, 31 May - 3 June 1999.
[19] David Buttler, Ling Liu, Calton Pu. "A Fully Automated Extraction System for the World Wide Web", IEEE ICDCS-21, Phoenix, Arizona, April 16-19, 2001.
[20] G.F. Luger, W.A. Stubblefield, "Artificial Intelligence: Structures and Strategies for Complex Problem Solving", Third Edition. Addison Wesley Longman, Inc., 1997.
[21] http://www.w3.org/People/Raggett/tidy/

최 훈 일
2000년 청주대학교 컴퓨터정보공학과 학사. 2000년 ~ 현재 청주대학교 전산정보공학과 석사과정. 관심분야는 HCI, VoiceXML, XML, 음성정보처리를 이용한 웹 프로그래밍

장 영 건
정보과학회논문지 : 컴퓨팅의 실제 제 7 권 제 5 호 참조