Voice 브라우저의 설계 및 구현

장준식*, 윤재석*
*대진대학교 컴퓨터공학과

Design and Implementation of VoiceXML Browser

Joonsik Jang*, Jaeseog Yoon*
*Dept of Computer Engineering, Daejin Univ.
E-mail: jsjang, jsyoon{@daejin.ac.kr}

요 약

본 연구에서는 기존의 전통적인 IVR 시스템이 갖는 제약을 해결할 수 있는 XML 포맷을 가지는 VoiceXML 문서를 브라우징할 수 있는 Voice 브라우저를 설계·구현하였다. VoiceXML로 기술된 다이얼로그를 VoiceXML 인터프리터를 통하여 해석하고 추출된 폼을 FIA로 해석하게 하였으며 음성 인식 엔진의 컴포넌트를 이용하여 Grammar 컴파일 및 음성 입·출력이 가능하도록 하였다. 본 연구의 브라우저를 기반으로 하는 시스템은 음성 언어 어플리케이션을 개발할 시에 음성 인식과 같은 복잡한 기술을 이용하지 않아도 되며 현재 웹 개발의 이점을 이용할 수 있다.

키워드
브라우저, VoiceXML, DOM

Ⅰ. 서 론

많은 사람들이 사용하는 음성 메일이나 전화 기반의 고객 지원 시스템, 텔레뱅킹 등은 대부분이 전통적인 IVR(Interactive Voice Response) 시스템으로 구성되어 있다. 이러한 독점적이고 폐쇄적인 시스템 구조를 공개적인 프로그래밍 가능한 아키텍처로 향상시키기 위한 연구들이 진행 중에 있다.

VoiceXML[3]은 전화용 어플리케이션을 위해서 합성된 음성, 디지털화된 오디오, 음성 인식과 DTMF 키 입력과 음성의 녹음의 기능을 가지는 음성 다이얼로그를 만들어내기 위하여 W3C[4]에 의해 개발된 XML 기반의 언어이다. VoiceXML은 HTML form이나 CGI 스크립트와 유사한 프로그래밍할 수 있는 다이얼로그을 제공함으로서 전화 사용자에게 웹기술의 이점을 가져다 주었으며 음성언어 시스템을 개발할 때 필요한 복잡한 기술 개발의 필요성을 많이 감소시킬 수 있다. 이 때문에 개발자들은 빠른 시스템의 프로토타입 및 음성 기반 어플리케이션을 개발할 수 있다[1][2].

본 연구에서는 XML 형식을 갖는 VoiceXML을 브라우징할 수 있는 브라우저를 DOM(Document Object Modal)[5]을 이용하여 설계/구현하였다. 입력된 VoiceXML을 XML 파서로 파싱하여 DOM 트리를 생성하고 이 트리 내에 form 엘리먼트 부분을 추출하여 FIA(Form Interpretation Algorithm)[3]에 적용하여 VoiceXML에 기술된 다이얼로그를 수행하게 함으로써 음성 브라우징이 가능하도록 하였다. 이때 음성 인식·출력·녹음을 Nuance System[8]의 컴포넌트를 이용하였다.

Ⅱ. DOM

DOM(Document Object Model)은 논리적인 문서의 구조와 문서에 접근하고 문서를 다룰 수 있는 방식을 정의한다. DOM은 플랫폼과 언어 독립적인 인터페이스이며 프로그램이나 스크립트들이 동적으로 문서에 접근하고 그 내용이나 구조, 스타일을 갱신할 수 있는 인터페이스를 제공하고 있다.

DOM으로 프로그래머가 문서를 만들고 구조를 조정하고 엘리먼트와 내용을 추가, 삭제, 수정을 할 수 있도록 설계되었다. DOM 인터페이스를 구

현하는 DOM 파서를 통해서 XML 문서가 파싱되면 DOM 인터페이스에 맞는 DOM 트리를 메모리에 생성한다. 이렇게 생성된 DOM 트리는 내부의 각 노드에 Random Access가 가능하고 구조 변경이 용이한 장점을 가지고 있다.

III. VoiceXML

VoiceXML은 AT&T, IBM, 모토롤라, 루슨트 테크놀로지 등에 의해 창립된 VoiceXML 포럼에서 제안된 대화 형 Markup 언어이다. 음성 애플리케이션 개발을 위해 고안된 XML 문서 형식의 일종으로 세계 인터넷 환경을 주도하고 있는 W3C 컨소시엄에서 VoiceXML 포럼의 제안을 받아들여 표준화되었으며 현재는 2.0버전까지 발표되었다.

VoiceXML은 기존의 전통적인 IVR 시스템이 가지는 독점적이고 폐쇄적인 단점들을 XML이라는 개방적인 스크립트 언어로써 해결하였다. 현재 많은 웹사이트의 개발에 쓰이는 웹을 기반으로 하는 개발 방법의 이점들을 가지고 올 수 있으며 IVR 시스템의 내용을 변경할 시에 발생하는 어려움도 상당 부문을 감소시킬 수 있다. 개발자들은 IVR 어플리케이션을 구현할 시에 음성 인식이나 음성 합성 등과 같은 개발이 어려운 부분에 대해서 잘 알고 있지 않더라도 음성 서비스를 구현할 수 있다.

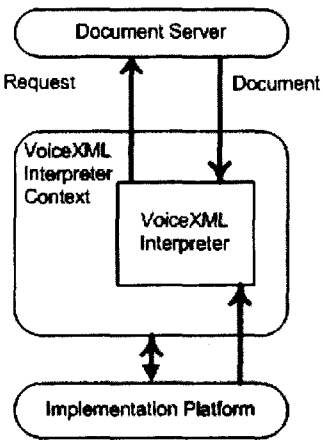

그림 1. VoiceXML 기반 시스템 아키텍처 모델

그림1은 VoiceXML을 기반으로 하는 시스템의 아키텍처 모델을 나타내고 있다. Document Server의 가장 일반적인 예로 웹서버를 들 수 있으며 VoiceXML 문서 및 오디오 파일, Grammar 파일 등의 저장소 역할을 한다. VoiceXML Interpreter Context는 VoiceXML Interpreter와 구현 플랫폼간을 이어주는 역할을 한다. VoiceXML 인터프리터는 VoiceXML 문서를 입력 받아 VoiceXML 내에 기술된 다이얼로그를 해석하는 역할을 담당한다. 이 때 해석된 내용에 따라서 사용자의 음성 인식을 수행하거나 다른 VoiceXML 문서로 이동하게 된다. 구현 플랫폼에는 음성 인식 엔진이나 음성 합성기 등이 위치하게 된다.

IV. DOM 기반 VoiceXML 인터프리터의 설계

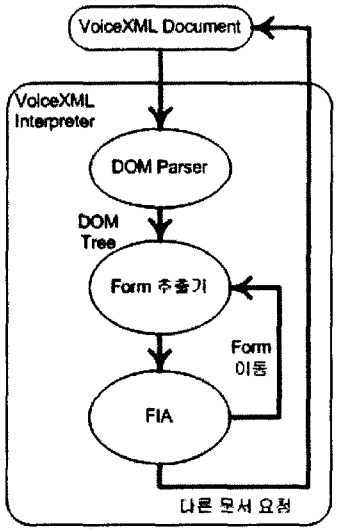

그림 2 VoiceXML 인터프리터의 작동

그림 2는 본 연구에서 설계·구현한 Voice 브라우저 내의 VoiceXML 인터프리터의 작동을 나타낸다. 인터프리터에 의해 요청된 VoiceXML 문서가 DOM 파서에 입력이 되면 파서는 문서의 정형성 및 유효성을 검사한 후에 DOM 인터페이스에서 정의된 타입의 노드로 DOM 트리를 메모리에 형성한다. Form 추출기는 DOM 트리를 순회하면서 form 별로 각 요소들을 별도의 데이터 객체로 변환한다. 이 form 데이터 객체는 id 속성이 가지는 값을 키로하여 별도의 데이터로 저장하여 추후에 id 값으로 form을 가져 올 수 있도록 하였다. Form 추출기를 통해서 추출된 form은 FIA Form Interpretation Algorithm)[3]에 맞도록 form 내부의 요소들을 인터프리트 하도록 하였다. FIA는 form을 선택하는 것과 form 내에 음성 인식과 같은 과정에 의해서 입력된 데이터를 저장 및 처리에 관한 알고리즘이다. 인터프리트 과정에서 다른 폼으로의 이동에 관련된 요소를 만나면 같은 문서 내의 폼일 경우에는 이전에 id 키로 저장했던 form을 꺼내와 FIA로 인터프리트를

하고 다른 문서일 경우에는 처음으로 돌아가 초기 과정을 반복하도록 하였다.

V. VoiceXML 브라우저 설계·구현

표 1. VoiceXML Browser 개발 환경

구 분	내 용
운영체제	윈도우2000 Pro
개발 툴	Eclipse, Java2 SDK 1.4
XML 파서	Apache Xerces-Java 2.5
음성인식엔진	Nuance ver. 8
문서 서버	Apache We Server 1.3

본 연구에서 설계·구현한 Voice 브라우저는 표 1과 같은 환경을 가진다. Pentium 4 1.5GHz PC에서 윈도우 2000 pro 운영체제를 기반 플랫폼으로 하였으며 주요 개발 툴로는 Java2 SDK 1.4[6] 버전의 Eclipse를 사용하였다. XML 문서를 파싱하고 DOM 인터페이스로 변환하는 XML 파서로 Apache Xerces-Java 2.5[7] 버전을 사용하였다. VoiceXML 브라우저의 중요한 기능인 음성 인식을 위하여 Nuance SRS(Speech Recognition System) 8 버전[8]을 사용하였다.

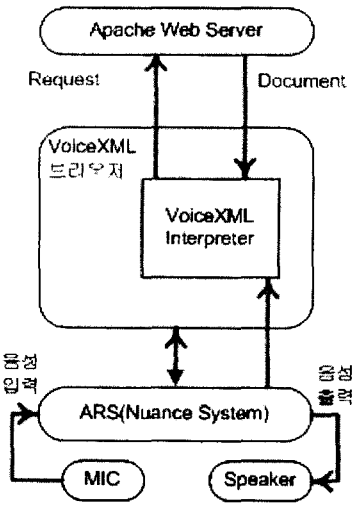

그림 3. VoiceXML 브라우저 구성도

그림3은 본 연구에서 설계·구현한 Voice 브라우저의 구성도를 나타낸다. Voice 브라우저는 외부로부터 VoiceXML 문서를 요청하고 이를 VoiceXML 인터프리터로 전송하는 역할을 하며 인터프리트를 진행하면서 실제 구현 플랫폼과 연동하여 음성 입·출력을 수행하게 하였다. 본 연구의 VoiceXML 브라우저에 사용된 VoiceXML 문서는 VoiceXML 2.0 스팩의 제한된 기능들을 수행할 수 있는데, form, field, prompt, audio, grammar, goto의 기능을 구현하였다.

VoiceXML 브라우저를 실행하면 프로그램에서 지정한 초기 VoiceXML문서를 문서 서버로부터 요청한다. 입력 받은 VoiceXML 문서는 XML 파서에 의해서 DOM 트리형태로 메모리에 적재되며 Form 추출기에 의해서 각 노드를 순회하며 form 요소를 찾아 별도의 데이터타입으로 저장하도록 하였다. 여기서 저장되는 form 노드는 DOM 인터페이스에 정의된 노드이므로 노드간 이동이 간단하다. 추출된 form은 VoiceXML 인터프리터에 의해서 FIA를 수행하게 하였다. form 내에는 그림4에서 보듯이 field나 prompt, grammar등의 요소들이 포함되어 있다. VoiceXML 인터프리터가 field 요소를 만나면 각 필드의 값을 저장할 데이터 공간을 만들며 prompt 요소를 만나면 구현 플랫폼에 오디오 리소스의 위치를 전달해주면 구현 플랫폼에서 스피커를 통해 오디오 파일을 재생한다. 본 연구에서 사용한 Nuance SRS는 음성 인식의 범위를 grammar내에 정의된 도메인으로 제한한다. 사용자가 말하는 임의의 음성을 바로 인식하는 것이 아니라 사용자가 입력할 데이터와 패턴을 정의하는 grammar를 해석하고 이 grammar에 정의된 내용만을 인식한다. 그림4와 같은 다이얼로그가 이루어지기 위해서는 grammar 파일에 도착지역과 시간 패턴을 정의해야 한다. 이 논문에서 사용한 grammar는 Nuance SRS의 고유 포맷인 GSL(Grammar Specification Language)[8]형식으로 작성하였다. 음성 인식을 수행한 후에 반환되는 결과값은 이전에 할당해 두었던 field에 (name,value) 쌍으로 저장된다. 이렇게 저장된 값들은 CGI와 같은 웹 프로그램에 전달하기 위한 파라미터 값으로 이용된다.

브라우저 : 목적지를 말씀해 주십시요
사용자 : 제주도
브라우저 : 몇 시에 출발하는 비행기를 예약하시겠습니까?
사용자 : 오후 3시

그림 4. 테스트 다이얼로그

그림4는 테스트에 사용된 다이얼로그의 한 부분으로 사용자가 비행기 예약을 하기 위해서 목적지 및 탑승 시간을 선택하는 다이얼로그이다. 이와 같은 다이얼로그를 수행하기 위해서 그림5와 같이 VoiceXML을 기술해야 한다. 사용자가 말하는 목적지 및 시간은 음성 인식을 위해서 GSL로 작성하였다. 그림5의 VoiceXML문서가

VoiceXML 브라우저에 입력되면 그림4와 같은 다이얼로그가 진행된다. 그림4와 같은 다이얼로그를 수행하는 음성 어플리케이션을 개발하기 위해서 전통적인 IVR 시스템들은 음성 인식 엔진과의 인터페이스를 하는 부분에 대한 개발의 부담이 있으며 다이얼로그에 맞는 시스템을 폐쇄적인 형태로 개발해야 했다. 그러나 본 연구의 Voice 브라우저의 경우에는 음성 어플리케이션을 구현하기 위해 그림5와 같이 XML로 다이얼로그를 기술함으로써 이전보다 간단하게 어플리케이션을 구현할 수 있다. 시스템 구현시에 어려움이 발생하는 음성 인식과 같은 부분은 이미 구현이 되어 있기 때문에, 실제 서비스의 내용인 그림5와 같은 VoiceXML 문서를 기술하는데 중점을 둘 수 있다. 이러한 XML 문서는 기술하기가 비교적 간단하고 추후에 변경이 용이하다는 장점을 가지고 있다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<!DOCTYPE vxml SYSTEM "vxml-2-0.dtd">
<vxml version="2.0" xml:lang="ko-KR">
<form id="movieselect">
  <field name="departure">
   <prompt><audio src="departure.wav"/>
   </prompt>
<grammar
        src="grammars/depature.grammar"/>
  </field>
  <block>
   <goto next="#time"/>
  </block>
</form>
<form id="time">
  <field name="time">
   <prompt>
     <audio src="prompts/time.wav"/>
   </prompt>
<grammar
        src="grammars/time.grammar"/>
  </field>
</form>
</vxml>
```

그림 5. 테스트에 사용한 VoiceXML

VI. 결 론

본 연구에서는 기존의 전통적인 IVR 시스템을 개선한 VoiceXML Browser를 설계·구현하였다. VoiceXML 브라우저는 음성 언어 어플리케이션을 개발할 때 복잡한 음성 인식 엔진에 대한 지식에 대해서 알고 있지 않더라도 시스템과 사용자간의 다이얼로그를 VoiceXML 스펙에 맞게 기술함으로써 음성 언어 어플리케이션을 구현할 수 있게 하였다. 이와 같은 브라우저는 JSP, PHP와 같은 웹 프로그래밍 기술을 이용하여 다이나믹한 VoiceXML 문서를 생성할 수 있는 이점을 이용할 수 있다.

VoiceXML 브라우저는 음성과 DTMF 신호의 입력와 음성 및 오디오 출력의 기능을 가지록 설계되었다. 이러한 시스템은 사용자의 상황에 따라서 제약을 가질 수도 있다. 차후에서는 VoiceXML과 HTML을 동시에 브라우징할 수 있고 Multi-Modal적인 접근이 가능한 시스템에 관한 연구가 필요하다.

참고문헌

[1] Kudan Singh, Ajay Nambi, Henning Schulzrinne, "Integrating VoiceXML with SIP services", Communications, 2003. ICC '03. IEEE International Conference on , Volume: 2 May 2003

[2] Bennett C, Font Llitjos A, "Building VoiceXML-based Applications" Proceedings of ICSLP 2002

[3] Voice Browser Activity, "http://www.w3.org/voice"

[4] W3C, "http://www.w3c.org"

[5] DOM, "http://www.w3.org/dom"

[6] Java SDK, "http://java.sun.com"

[7] Apache Xerces, "http://xml.apache.org"

[8] Nuance SRS, "http://extranet.nuance.com"