웹 기반의 VoiceXML 문서 인터프리터의 설계

이선남* 김경아 이기호
이화여자대학교 컴퓨터학과
{sunnyday, kakim,khlee}@mm.ewha.ac.kr

The design of VoiceXML Interpreter based on the Web

Sun-Nam Lee* Kyung-Ah Kim Ki-Ho Lee
Dept. of Computer Science & Engineering, Ewha Womans University

요 약

VoiceXML은 음성인식 및 음성합성과 같은 음성처리기술을 이용하여, 시각에 의존하는 기존의 웹을 벗어나 음성 및 시각을 모두 활용할 수 있는 새로운 정보 서비스 패러다임으로 제시되어지고 있다. VoiceXML을 이용한 음성정보서비스를 제공할 경우, 마크업 언어형태로 작성된 시나리오를 인터프리터를 통해 서비스하기 때문에 시나리오 변경 요구시 재프로그램해야 하는 기존 음성정보서비스 시스템의 문제점을 쉽게 개선할 뿐만 아니라, 음성정보서비스의 개발자가 음성인식 · 음성합성과 같은 기술적인 문제와는 독립적으로 시나리오를 작성할 수 있다는 이점이 있다.

본 논문에서는 W3C Voice Browser Working Group에서 제안하는 문법표현 · 시스템구조 · 다이얼로그 모델 등을 지원하는 XML 기반 대화형 마크업 언어인 VoiceXML 문서의 인터프리터를 설계하고자 한다.

1. 서론

기존 음성 서비스 시스템의 경우, 개발자가 음성인식 · 음성합성 · 자연어처리 기술들을 모두 이해해야만 프로그램이 가능하며, 시나리오 변경 요구시 음성 재녹음을 비롯하여 프로그램을 재구현하는 등 개발 시간과 개발 비용이 매우 높다는 제약이 따랐다. 웹 서비스 또한 Display, Keyboards, Pointing Device를 통한 시각 위주의 것이 주를 이루어 왔다. 이러한 시각 위주의 웹 서비스에서 벗어나, 음성인식과 합성기술을 기반으로하여 시각과 음성을 모두 활용할 수 있는 새로운 패러다임이 제시되어 지고 있다.

이러한 시점에서 AT&T, IBM, Lucent Technologies, Motorola 등이 설립한 VoiecXML 포럼에서는 음성 어플리케이션 개발을 위한 XML 기반 언어로서 VoiceXML(Voice eXtensible Markup Language)을 제안했으며, W3C(World Wide Web Consortium)에서 이 제안을 받아들여 웹의 대화형 마크업 언어로 VoiceXML을 표준으로 공인했다. 이에 음성 서비스 시나리오 작성자는 음성 입출력의 기술적인 문제에 대한 지식 없이도 VoiceXML 문서를 가지고 음성포털을 구축할 수 있게 되었다.

본 논문에서는 사용자와 컴퓨터간에 서로 대화할 수 있는 음성 어플리케이션을 위한 XML 기반의 대화형 마크업 언어인 VoiceXML 문서의 Interpreter를 설계하고자 한다.

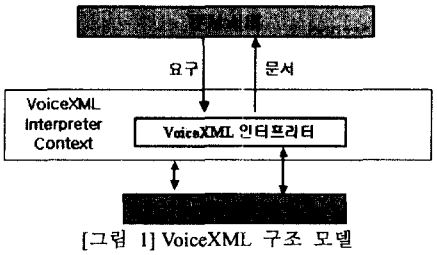
2. VoiceXML

VoiceXML은 음성 입출력이 가능한 대화형 음성서비스를 위한 표준언어로서, 프로그래머로 하여금 저수준 언어로 프로그래밍해야하는 기존의 어려움과, 자원을 일일이 관리해야 하는 문제점으로부터 벗어나게하고, 서버/클라이언트 환경에서 데이터 서비스와 음성 서비스를 통합시키는 것을 가능하게 하였다.

본 논문에서는 Voice Browser Working Group에서 제안한 시스템 구조, 문서 구조, 다이얼로그 모델, 문법 표현 등을 지원하는 인터프리터를 설계한다.

2.1 구조 모델

VoiceXML 실행환경은 [그림1]과 같은 구조를 갖는다. VoiceXML 인터프리터는 VoiceXML 문서를 적재하고 그 내용을 해석해 실행하는 역할을 한다. 즉 다이얼로그, 문법, 이벤트, 오디오출력, 오디오입력, 콜 제어, 흐름 제어와 관련된 47종의 각 태그에 설정된 기능에 따라 문서 실행의 순차적 흐름을 제어하고, 음성 입출력 내용을 결정해 음성 플랫폼에 필요한 명령을 내린다. VoiceXML Interpreter Context는 인터프리터를 통제·관리하며, 문서 서버는 URI(Uniform Resource Identifier)형태로 요청되어지는 VoiceXML 문서·오디오 파일·grammar 파일 등을 인터프리터에게 전송하는 역할을 한다.



[그림 1] VoiceXML 구조 모델

2.2 문서 구조

VoiceXML 문서 형태는 시나리오 구성형태에 따라 다음과 같은 세가지로 분류된다.

(1) 단일 문서(Single document application) : 단일 문서로 서비스 시나리오를 구성한 것이다.

(2) 멀티 문서(Multi document application) : 여러 개의 하위 문서와 하나의 root 문서로 서비스 시나리오를 구성한 것으로서, root 문서에서 정의한 변수, 문법, 다이얼로그 정보를 하위문서가 사용할 수 있다.

(3) 서브 다이얼로그(Subdialog) : 자주 사용하는 다이얼로그를 모듈화하여 재사용 가능하게 한다.

2.3 다이얼로그 모델

VoiceXML 다이얼로그는 다이얼로그 진행 변화 유무에 따라, 다음과 같이 두가지로 분류된다.

(1) Computer directed form : 미리 정의된 순서에 따라 사용자와 컴퓨터가 대화하는 형식을 가지고 있는 수동적인 다이얼로그 모델이다.

(2) Mixed initiative form : 사용자와 컴퓨터 모두 대화의 진행을 능동적으로 변경할 수 있는 다이얼로그 모델로서, 사용자의 입력에 따라 다른 다이얼로그가 실행된다

2.4 문법 형태

Voice Browser Working Group에서는 음성인식에 필요한 정보를 제공하기 위하여, Grammar라는 인식단어 집합을 정의하였으며, 이를 위해 문법 형태를 표준화 하였다. VoiceXML의 문법 파일 포맷으로는 JSGF(Java Speech Grammar Format)를 사용하며, JSGF는 단어집합을 문맥 무관 문법(Context Free Grammar)으로 표현하여 다양한 형태의 문장 및 단어 집합을 생성 가능하게 하였다.

2.5 엘리먼트 분류

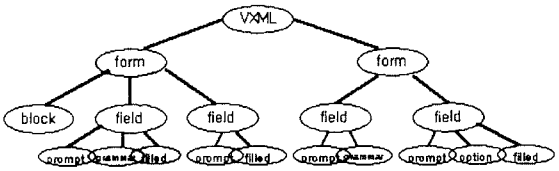
47개의 Voicexml 엘리먼트에 대한 Action을 정의하기 위해 [표1]과 같이 10개의 카테고리로 엘리먼트를 분류하여 분석하였다.

[표1] VoiceXML 엘리먼트의 분류

root	vxml, meta
dialogues	form, menu, choice
prompt	prompt, enumerate, reprompt
fields	field, option, var, initial, block, assign, clear, value
events	catch, error, help, link, noinput, nomatch, throw
audio output	audio, break, div, emp, pros, sayas
audio input	dtmf, grammar, record
call control	disconnect, transfer
control flow	if, elseif, else, exit, filled, goto, param, return, subdialog, submit
miscellaneous	object, property, script

3. VoiceXML Interpreter

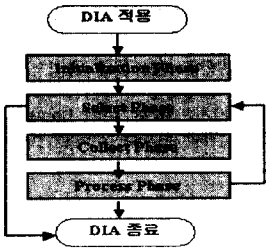
VoiceXML 인터프리터는 XML 문서 모델링을 위해 정의된 DOM(Document Object Model) 트리 형태와 같이, 문서의 엘리먼트를 트리의 노드로 가지는 문서구조로 표현하며, 트리의 노드를 순회하면서 해석한다. 트리를 순회할 때, BFS(breadth-first search) 방식과 DFS(depth-first search) 방식을 혼합한 Hybrid 방식으로 순회하도록 하였다. 즉 DFS 방식을 우선으로 하되 특정 엘리먼트의 요구가 충족된 경우, 특정 엘리먼트의 이웃노드를 순회하는 BFS 방식을 사용하였다. [그림2]의 경우, VXML 엘리먼트를 수행한 다음, DFS 방식에 의해서 왼쪽 노드의 Form 엘리먼트를 수행해야되지만, 이미 다른 문서나 다이얼로그에 의해 Form 엘리먼트의 변수가 지정되어 있다면, 그 아래의 child 노드들을 무시하고, BFS 방식에 의해 이웃노드인 Form 엘리먼트를 수행하도록 하였다.



[그림 2] VoiceXML 문서의 트리 예

3.1 Dialog Interpretation Algorithm (DIA)

VoiceXML 문서는 대화형 음성서비스를 위한 마크업 언어이기 때문에 다이얼로그 위주로 문서가 구성되어지며, 이를 위해서 다이얼로그 해석 알고리즘을 작성하였다. DIA 알고리즘은 Initializaion 단계, Select 단계, Collect 단계, 그리고 Process 단계로 구성되어지며, 이 알고리즘은 인터프리터의 주요 해석 알고리즘이며 해석되어지는 방식은 다음과 같다.



[그림 3] DIA 알고리즘

3.1.1 Initialization Phase

Form 엘리먼트를 시작하면서 Form과 Form Item의 변수값을 초기화하는 단계로서 다음을 실행한다.

i. 변수를 선언하고 form item 변수의 expr 속성값 초기화.
ii. Field item인 경우, 프롬프트 카운터를 1로 셋팅.
iii. Initial item이 있다면, 프롬프트 카운터를 1로 셋팅.

3.1.2 Select Phase

다음에 실행할 Form Item을 선택하는 단계로서 다음을 실행한다.

i. 실행할 아이템이 지정된 경우, 명시된 Form Item을 선택
ii. 명시되지 않았을 경우, 정의되지 않은 가드 변수를 가진 Form Item 중에서 첫번째 Form Item을 선택.
iii. 더 이상 실행할 Form Item이 없을 경우, DIA를 종료.

3.1.3 Collect Phase

Prompt를 실행한 후, 현재의 Form Item에 적용될 문법을 활성화시킨 다음, 사용자의 입력이나 이벤트를 기다리는 단계로서 다음을 수행한다.

i. <prompt> 를 선택한 후, 사용자에게 프롬프트 하고, 프롬프트 카운터를 1만큼 증가.
ii. 현재의 폼 아이템에 적용될 문법 활성화.
폼 아이템의 modal 속성을 검사하여, " true" 이면 field 아이템 만을 위한 문법 초기화 및 인식 사전을 구축.
false" 인 경우, 현재의 인식 사전에 field 아이템 수준의 문법을 추가하여 인식 사전을 구축.

iii. Form Item을 실행.
(i) Form Item = Field, initial 라면, 사용자의 입력이나 이벤트를 기다림.
(ii) Form Item = object 라면, oobject 실행, object 의 변수를 반환되는 ECMA script 값으로 셋팅.
(iii) Form Item = subdialog 라면, subdialog 실행, subdialog의 변수를 반환되는 ECMA script 값으로 셋팅.
(iv) Form Item = transfer 라면, transfer. 만약 wait=true라면 transfer의 변수를 반환되는 상태로 셋팅.
(v) Form Item = block 라면, block의 변수를 정의된 값으로 셋팅, block의 context 수행.

3.1.4 Process Phase

Collect Phase에서 모은 사용자의 입력이나 이벤트를 처리하는 단계로 다음을 수행한다.

i. 사용자의 입력이 이벤트에 해당하는 경우, 이벤트 핸들링. 사용자가 말을 하지않거나(noinput), 잘못된 입력(nomatch) 등의 특정 이벤트가 발생한 경우, catch를 찾아 실행.
ii. 사용자의 입력이 form 외부의 grammar로 인식되는 경우.
(i) grammar가 link 엘리먼트에 속해 있을 경우, link의 goto 나 throw를 실행 , DIA 종료
(ii) grammar가 menu의 choice 엘리먼트에 속해 있을 경우, choice의 goto나 throw 실행 , DIA 종료
(iii) 만약 grammar가 다른 form에 속해 있을 경우, 다른 form의 DIA로 입력된 음성을 보냄.
iii. 사용자의 입력이 form내부의 grammar로 인식되는 경우
(i) 입력된 음성을 Form Item 변수로 copy.
(ii) 만약 어떤 field 아이템이라도 채워져있다면, <initial> form 아이템 변수를 true로 셋팅.
(iii) 입력된 음성에 의해 행해지게 되는 filled 수행.
filled가 form item의 자식 노드일 경우, Filled_변수=form 아이템 변수의 값이고, Filled가 Form의 자식 노드일 경우, Filled_변수 = 그 form에 속해 있는 변수의 값들이다. 변수를 저장하고 Filled action을 취한다.

위의 DIA 알고리즘에 의해서 [문서1]을 해석해보면, 다음과 같은 시나리오로 해석되어진다.

```
컴퓨터 : VocieXML 날씨정보 서비스입니다.
         어떤 지역의 날씨정보를 원하십니까?
         서울, 대전, 대구, 부산, 제주, 기타 에서 선택하십시오
사용자 : ....
컴퓨터 : 원하시는 지역을 말씀하십시오
사용자 : 서울
컴퓨터 : 현재 서울지역 기온은 20도이며, 습도는 20% 입니다.
```

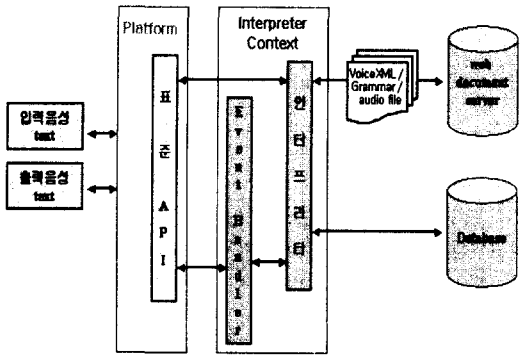
4. VoiceXML Interpreter 실행구성도

VoiceXML 인터프리터의 실행구성도는 [그림 4]와 같다.

VoiceXML 인터프리터는 web document server로부터 VoiceXML 문서를 적재하여 문서의 내용을 47개의 태그 특성과 DIA 알고리즘에 의하여 해석하고 실행한다. 인터프리터는 음성엔진을 결합시키지 않았기 때문에 음성출력 엘리먼트를 수행해야 할 경우, 출력음성 내용을 표준 API를 통해 text로 내보내고, 입력음성을 text로 받았을 경우에는 입력된 음성을 DIA 알고리즘에 의하여 단어인식사전에서 검색하여, 다음 엘리먼트를 수행하는 식으로 진행된다. 만약 사용자의 입력이 이벤트에 해당하는 경우에는 이벤트 핸들러에게 이벤트 정보를 보내주어 핸들링하도록 한다. 또한 web document server는 URI 형태로 파일을 전송할 수 있도록 설계되었다.

```
<?xml version=" 1.0" ?>
<vxml version=" 1.0" >
<form id=" wea_info" >
<block> VocieXML 날씨정보 서비스입니다. </block>
<field name=" wea_area" >
<prompt> 어떤 지역의 날씨정보를 원하십니까?
서울,대전,대구,부산,제주,기타에서 선택하십시오
</prompt>
<grammar>서울/대전/대구/부산/제주/기타</grammar>
<catch event=" help" >원하시는 지역을 말씀하십시오
</catch>
<filled>
<if cond=" wea_area == '서울'" >
<prompt> 현재 서울지역 기온은 20도이며, 습도는 20% 입니다 </prompt>
</if>
</filled>
</field>
</form>
</vxml>
```

[문서 1] VoiceXML 문서의 예



[그림 4] 인터프리터 실행구성도

5. 결론 및 향후 연구

본 논문에서는 사용자와 컴퓨터간에 상호 interactive하게 대화할 수 있는 대화형 마크업 언어인 VoiceXML 문서를 위한 DIA 알고리즘을 소개하고 인터프리터를 설계하였다.

VoiceXML의 최종 목적은 웹이나 전화 기반의 음성서비스이기 때문에, 음성엔진을 사용하여 웹 기반의 음성인식과 음성합성을 지원하는 대화형 브라우저를 설계하고 구현하는 것을 향후 과제로 남기고자 한다.

[참고 문헌]

[1]Bruce Lucas, "VoiceXML for Web-based Distributed Conversational Applications", Communications of the ACM, 2000
[2]Thomas Ball, Veta Bonnewell, Peter Danielsen, "Speech-Enabled Services Using Teleportal Software and VoiceXML", Bell Labs journal, 2000
[3]Stephen Arnold, Leo Mark, John Goldthwaite, "Programming by Voice, VocalProgramming", ACM, 2000
[4]VoiceXML forum, Voice eXtensible Markup Language, http://www.voicexml.org/specs/VoiceXML-100.pdf
[5]World Wide Web Consortium Voice Browser Working Group, http://www.w3c.org/Voice
[6]김학균, "대화형 음성언어 인터페이스를 위한 VXML 인터프리터 개발", 음성통신 및 신호처리 학술대회 논문집, 2000