SALT 기반 음성 브라우저의 설계 및 구현

이용희° 이동우 신희숙 최은정 박준석
한국전자통신연구원 디지털홈연구단 차세대PC그룹
{lyhcool°, hermes, hsshin8, ejchoi, parkjs}@etri.re.kr

Design and Implementation of SALT-based Voice Browser

YongHee Lee°, DongWoo Lee, HeeSook Shin, EunJeong Choi, JunSeok Park
Smart Interface Research Team, Post-PC Research Group, ETRI

요 약

정보통신 기기의 발전하면서 소형화, 경량화와 함께 이동성을 갖춘 다양한 차세대 PC 기기들이 나타나고 있다. 기존의 마우스나 키보드를 통한 인터페이스뿐만 아니라 음성, 펜, 제스처 등을 이용한 멀티모달 인터페이스에 대한 요구가 증대되면서 이에 대한 연구가 활발히 이루어지고 있다. 또한 최근의 음성 처리 기술이 발전하고 단말기의 성능이 개선되면서 음성을 이용한 인터페이스에 대한 연구가 활발히 이루어지고 있다. 본 논문에서는 브라우저에서 음성 지원을 위해 제안된 SALT를 기반으로 하여 사용자와 음성 인터페이스가 가능한 음성 브라우저를 설계하고 구현한다.

1. 서 론

최근의 정보통신 기술의 발달과 더불어 PDA, 스마트폰 그리고 태블릿 PC 등과 같이 소형화, 경량화 그리고 이동성을 갖춘 다양한 기기의 보급화가 빠르게 진행되고 있다. 이러한 차세대 PC 기기들의 등장과 함께 사용자들과의 밀착된 관계를 유지하기 위해 입력과 출력의 불편을 해소하기 위한 방안으로 멀티모달 인터페이스에 대한 요구가 증대되고 있다. 멀티모달 인터페이스란 인간과 컴퓨터와의 상호 의사소통을 위한 정보 입력과 출력을 위해서 다양한 방법을 사용하여 인터페이스를 제공하는 것을 말한다. 이러한 것은 마이크, 펜, 카메라 등의 인터페이스를 사용하여 입력된 음성, 문자 그리고 영상 정보를 각각의 사용 환경과 용도에 따라 이용하여 다양한 기술 활용이 가능하게 만든다.

이러한 멀티모달 인터페이스를 지원하기 위해 표준 스펙들을 작성하는 작업이 활발히 진행되고 있다. 그 중에서 음성 인터페이스를 위해 만들어지고 있는 대표적인 스펙들은 VoiceXML[1][7][8][9][10], X+V (XHTML+Voice) [2] 그리고 SALT (Speech Application Language Tags) [3][11][12]가 있다. VoiceXML은 전화를 통해 웹 서버에 접속하는 환경을 위해 만들어진 스펙이며 X+V는 기존의 XHTML과 VoiceXML을 적절하게 조합하여 브라우저에서 음성 인터페이스가 가능하도록 설계된 스펙이다. SALT 역시 브라우저에서 음성 인터페이스를 위한 스펙이다. 본 논문에서는 SALT 스펙을 이용하여 브라우저에서 음성 인터페이스가 이루어지도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고, 3장과 4장에서는 본 논문에서 설계하고 구현한 음성 브라우저에 대하여 설명하고, 5장에서는 본 논문의 결론 및 향후 과제에 대해 기술한다.

2. 관련연구

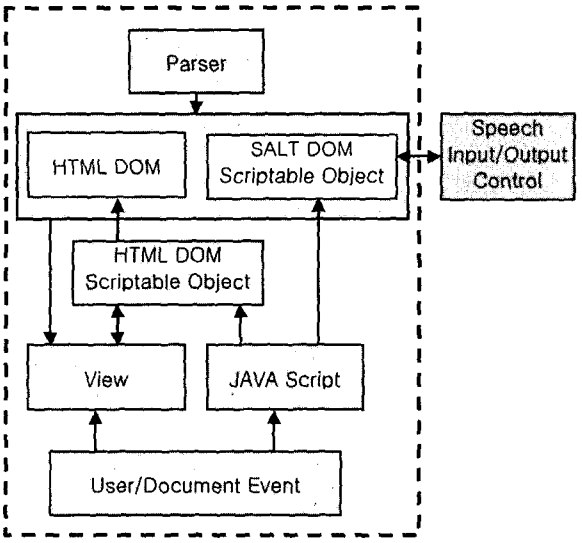
SALT(Speech Application Language Tags)는 W3C가 주도하는 VoiceXML과 달리 마이크로소프트가 주도하고 시스코, 컨버스, 인텔, 필립스, 스캔소프트, 스피치웍스 등이 참여하여 만든 SALT 포럼에서 만든 스펙이다. SALT 포럼은 멀티모달 인터페이스를 위한 음성 기술의 표준화를 추진하고 있다. 2001년 설립된 SALT 포럼은 전세계 70개 이상의 기업이 참여하고 있으며 다양한 종류의 정보기기 (PC, 전화, Tablet PC, PDA)에서 음성으로 여러 컨텐츠에 접근할 수 있도록 지원하기 위한 표준을 제정하고 시행하기 위해 설립됐다.

SALT는 VoiceXML처럼 하나의 언어 풀셋을 정의하지 않고 PC용 비주얼 웹 브라우저를 위한 HTML[4] 또는 XHTML[5]이나, 휴대전화 또는 PDA용 웹 브라우저를 위한 WML[6]에 내포될 수 있도록 설계되었다. 웹 어플리케이션에 음성 인터페이스용 태그를 추가할 수 있어 마우스나 키보드뿐만 아니라 음성 입력을 통하여 어플리케이션의 제어가 가능하다. 마이크로소프트사는 여러 제품군에 SALT를 적용하고 있으며 닷넷 환경에서도 SALT를 사용할 수 있도록 지원하고 있다.

SALT 1.0 스펙은 음성 지원을 위한 4개의 최상위 엘리먼트 prompt, listen, dtmf, smex와 2개의 오브젝트 PromptQueue, CallControl Object로 구성되어 있다. 4개의 최상의 엘리먼트에서 prompt 엘리먼트는 음성 합성 및 출력을 위한 것이고 listen 엘리먼트는 음성 인식 및 입력을 위한 엘리먼트이다. dtmf 엘리먼트는 전화 다이얼링을 위한 것이고 smex 엘리먼트는 시스템 플렛폼과 메시지를 주고 받고 또한 디버깅 메시지 등을 지원하기 위한 엘리먼트이다. 그리고 2개의 오브젝트에서 PromptQueue Object는 prompt 객체들의 재생 기능을 제어하기 위한 것이고 CallControl Object는 전화 다이얼링을 위한 SALT 프로파일에 사용하기 위한 오브젝트이다. 이 중에서 브라우저의 음성지원을 위해 prompt, listen 엘리먼트 그리고 PromptQueue Object를 이용하여 구현한다.

3. 브라우저 설계

현재 기본적인 컴퓨터의 입력 수단인 키보드와 마우스와 함께 SALT 기반으로 만들어진 웹 문서를 가지고 음성의 입력과 출력이 가능한 브라우저의 구조는 아래 [그림 1]과 같다.



[그림 1] SALT 지원 음성 브라우저 구조

HTML+SALT 파서는 SALT를 이용한 웹 문서를 가져와 파싱하는데 시각적인 인터페이스를 위한 HTML과 음성 인터페이스를 위한 SALT가 결합된 형태의 웹 문서를 분석하여 브라우저에서 필요로 하는 DOM을 만들어 낸다. 이때 생성한 DOM은 음성 입출력 엔진인 ASR(Automatic Speech Recognition)과 TTS(Text To Speech) 서버와의 인터페이스를 제공한다.

4. 브라우저 구현

4.1 음성 합성 및 출력

브라우저는 SALT 스펙을 따르는 다큐먼트의 텍스트 컨텐츠에 대한 음성 합성과 출력을 제공하고 오디오 컨텐츠에 대한 음성 출력 기능을 제공한다. SALT 스펙의 Prompt, Content, Value, Param 엘리먼트로 음성 출력 대상을 표현하고 PromptQueue Object로 제어한다.

브라우저의 음성 합성 및 출력 컴포넌트에서는 음성 합성에 필요한 정보를 추출하고 PromptQueue Object를 통하여 음성 출력을 제어 및 관리한다. 그리고 실제 음성 합성은 TTS 서버를 통하여 이루어지고 음성 출력은 서버와의 API를 통하여 이루어진다. JAVA로 구현한 브라우저 모듈과 TTS 서버와의 통신 API를 맞추기 위해서 JNI Native 코드를 사용하였고, 브라우저에서는 음성 합성 정보를 담기 위한 구조체를 정의하고 이 구조체에 맞도록 필요로 하는 정보를 TTS 서버에 전달한다.

4.2 음성 인식 및 입력

브라우저는 SALT 스펙을 따르는 다큐먼트 컨텐츠에 대한 음성 인식과 입력 기능을 제공한다. SALT 스펙의 Listen, Grammar, Record, Bind, Param 엘리먼트로 음성 인식 대상을 표현하고 제어한다.

브라우저의 음성 인식 및 입력 컴포넌트에서는 음성 인식에 필요한 정보를 추출한다. 추출한 정보는 ASR 서버와의 API를 통하여 보내어져 음성 인식 과정이 이루어진다. ASR 서버에서 음성 인식이 수행되고 인식 결과는 서버와의 API를 통하여 브라우저로 보내지게 된다. 음성 합성과 마찬가지로 음성 인식에서도 JAVA로 구현한 브라우저 모듈과 ASR 서버와의 통신 API를 맞추기 위해서 JNI Native 코드를 사용하였고, 브라우저에서는 음성 인식 정보를 담기 위한 구조체를 정의하고 이 구조체에 맞도록 필요로 하는 정보를 ASR 서버에 전달한다. 그리고 서버에서 음성 인식 결과를 브라우저에 보낼 때 음성 인식 결과와 함께 음성 녹음된 파일의 위치, 파일의 형식, 파일의 크기 그리고 음성 녹음 길이에 대한 정보를 함께 보내어 다른 응용 서비스가 가능하도록 하였다.

4.3 실험 예제

구현한 브라우저의 테스트를 위해 피자 주문 테스트 페이지를 작성하였다. SALT 스펙을 준수하는 테스트 페이지를 작성하여 브라우저에서 음성 입력과 출력이 제대로 이루어지는지 테스트 하였다.

```
<salt:prompt id="pmenu" oncomplete="lmenu.Start()">
피자의 종류를 선택해 주십시오. 고구마 피자, 불고기 피자, 치즈 피자가 있습니다.
</salt:prompt>
<salt:prompt id="psize" oncomplete="lsize.Start()">
피자의 크기를 선택해 주십시오. 크기는 대, 중, 소가 있습니다.
</salt:prompt>
```

[그림 2] 음성 출력을 위한 소스 코드

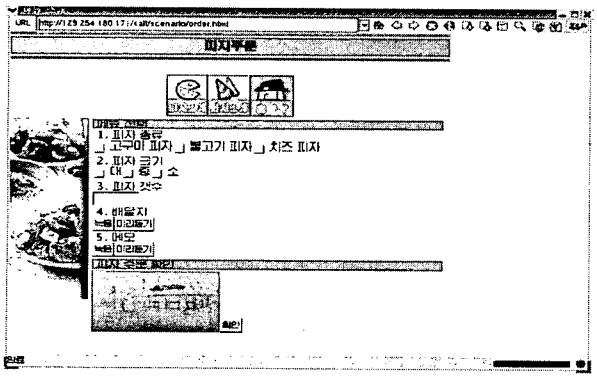
[그림 2]는 SALT 스펙을 준수하는 테스트 페이지에서 음성 출력을 위한 소스 코드의 일부분이다. 피자 주문을 위해 피자의 종류, 크기 그리고 수량 등을 음성 출력을

통하여 사용자에게 알려준다. 음성 출력에 사용될 문장들은 prompt 엘리먼트를 사용하여 작성된다. 사용자는 음성으로 출력되는 메뉴를 듣고 자신이 원하는 메뉴를 직접 음성으로 입력하거나 마우스 등의 다른 입력 장치를 이용하여 원하는 메뉴를 선택한다.

```
<salt:listen id="lmenu" onreco="menuResult()" onerror="pmenu.
Start()"> <salt:grammar>
고구마피자,; 불고기피자,; 치즈피자,;
</salt:grammar> </salt:listen>
<salt:listen id="lsize" onreco="sizeResult()" onerror="prompt_
size_again()"><salt:grammar>
대,; 중,; 소,;
</salt:grammar></salt:listen>
```

[그림 3] 음성 인식을 위한 소스 코드

[그림 3]은 음성 인식 대상 단어들을 음성 인식 서버에 전해주기 위한 소스 코드의 일부분이다. 본 연구에서는 음성 인식 서버로 사용한 모델이 SALT 스펙을 완전히 준수하지 못하여 테스트에 사용된 음성 인식 서버의 문법에 따라 페이지를 구성하였다. 그러나 SALT 스펙을 준수하는 음성 인식 서버를 사용할 경우에도 문법만 맞추어 페이지를 구성하면 서비스를 제공하는데 어려움이 없도록 구현되어 있다. 음성 인식을 위한 단어 규칙은 grammar 엘리먼트를 사용하여 정의한다.



[그림 4] 피자 주문 테스트 페이지

[그림 4]는 구현한 브라우저에서 피자 주문을 위한 테스트 페이지이다. 일반 웹 브라우저에서 볼 수 있는 웹 페이지의 모습과 동일하다. 하지만 음성을 이용하여 브라우저와 사용자 사이에 인터페이스가 이루어진다는 것이 다른 점이다. 각각의 단계에 따라 피자 주문을 할 수 있도록 음성 입력과 출력이 이루어진다. 이러한 피자 주문 페이지뿐만 아니라 전체 페이지를 구성할 때 음성 인터페이스를 추가하면 사용자는 더욱 편리하게 서비스를 이용할 수 있다.

5. 결론

본 논문에서는 브라우저에서 사용자와의 음성 인터페이스가 가능하도록 음성 지원을 위해 제안된 SALT 스펙을 기반으로 음성 브라우저를 설계하고 구현하였다. 사용자는 기존의 키보드와 마우스를 통하여 브라우저를 사용할 뿐만 아니라 음성 인터페이스를 통하여 더욱 편리하게 브라우저를 사용할 수 있다.

사용자와 컴퓨터 사이의 인터페이스에 대한 연구는 컴퓨터가 발명되었을 때부터 진행되어 왔다. 정보통신 기술의 급속한 발전은 사용자에게 더욱 새롭고 편리한 인터페이스를 요구하게 하였다. 그에 따라 인터페이스에 대한 연구가 더욱 활발하게 이루어지게 하고 있다. 본 연구를 통해 고성능, 초소형화와 더불어 이동성과 편의성이 향상되고 있는 차세대 PC 사용 환경에서 음성 인터페이스 활용의 기반이 될 것으로 기대된다.

6. 참고 문헌

[1] W3C, Voice Extensible Markup Language (VoiceXML) Version 2.0, Recommendation, Mar. 2004.

[2] VoiceXML Forum, XHTML+Voice Profile 1.2, Mar. 2004. http://www.voicexml.org

[3] SALT Forum, Speech Applications Language Tags (SALT) 1.0, Jul. 2002. http://www.saltforum.org

[4] W3C, HyperText Markup Language (HTML) 4.0, Recommendation, Apr. 1998.

[5] W3C, Extensible HyperText Markup Language (XHTML) 1.0, Recommendation, Jan. 2000.

[6] WAP Forum, Wireless Markup Language (WML), Apr. 1998. http://www.wapforum.com

[7] Niklfeld, G., Finan, R. and Pucher, M., "Architecture for Adaptive Multimodal Dialog System Based on VoiceXML", EuroSpeech 2001, pp.2341-2344, Aalborg, Denmark, Sep. 2001.

[8] Teppo, A. and Vuorimaa, P. "Speech Interface Implementation for XML Browser", ICAD 2001, Espoon, Finland, Jul. 2001.

[9] Eberman, B., Carter, J., Meyer, D., Goddeau, D. "Building VoiceXML Browsers with OpenVXI", WWW 2002, pp.713-717, Honolulu, Hawaii, May. 2002.

[10] Carpenter, B., Caskey, S., Dayanidhi, K., Drouin, C. and R. Pieraccini, "A Portable, Server-Side Dialog Framework for VoiceXML", ICSLP 2002, pp.2705-2708, Denver, Colorado, Sep. 2002.

[11] Wang, K., "SALT: a spoken language interface for web-based multimodal dialog systems", ICSLP 2002, pp.2241-2244, Denver, Colorado, Sep. 2002.

[12] Wang, K., "Semantic Object Synchronous Understanding in SALT for Highly Interactive User Interface", EuroSpeech 2003, pp.1209-1212, Geneva, Switzerland, Sep. 2003.