VoiceXML을 사용한 상가 검색 음성인식 시스템의 설계 및 구현

김우일*, 송성균*, 고경만*, 윤재석*, 김국보*
*대진대학교 컴퓨터공학과

Design and Implementation of Store Locator Voice Recognition System Using VoiceXML

Woo-Il Kim*, Sunggyun Song*, Kyungman Ko*, Jaeseog Yoon*, Gukboh Kim*
*Dept. of Computer Engineering, Daejin University
E-mail : groundp@korea.com

요 약

음성은 컴퓨터와 인간 사이의 인터페이스로서 지속적인 연구가 되어 왔다. VoiceXML로 구현된 음성 포털 서비스는 사용자의 음성 질의에 따라 정보를 검색하고 청취할 수 있는 기술로서 현재 다양한 컨텐츠로 서비스가 진행되고 있다. 본 연구에서는 전화나 인터넷 전화 프로그램으로 상가의 위치, 전화 번호, 상가 소개 등의 정보를 음성으로 검색할 수 있는 시스템을 VoiceXML을 이용하여 구현하여 보았다. 웹과 연동할 수 있도록 시스템을 구성하고 다양한 다이얼로그를 표현하기 위해 특히, JSP를 이용하고 각 로직을 자바빈즈 컴포넌트로 구현하였다.

1. 서 론

음성은 인간의 통신 수단 중에서 가장 자연스럽고 보편적인 의사 전달 수단이다. 현재의 음성 인식 기술은 매우 다양한 응용분야를 가지며 음성과학과 컴퓨터기술의 발전에 의해 크게 발전되어오고 있다. 기계와 인간의 통신을 위해 단어와 문장을 인식하는 음성인식(ASR, Automatic Speech Recognition), 음성합성(TTS,Text-To-Speech),화자인증(Speaker Verification) 을 기반으로 하는 효과적인 음성포털 서비스의 구현에 대한 활발한 연구와 개발이 이루어지고 있다.

컴퓨터형태의 발전 추이가 저용량 이면서 휴대 기능으로 변화함에 따라 과거 음성 인식의 주를 이루어 왔던 PC 환경 하 에서 뿐만 아니라, PDA, 이동 통신 단말기 등의 모바일 환경에서 Web 컨텐츠에 접근 활용할 수 있는 최적화된 음성 인식 기술이 필요하게 되었다. 음성인식 기술이 실용화 수준으로 발전함에 따라 음성 포탈을 비롯한 음성인식 응용 분야가 새로운 이슈로 떠오르고 있다. Web 컨텐츠에 접근 활용할 수 있는 최적화된 음성 인식 기술이 필요하게 되었다. 이러한 기존의 마우스나 자판 입력을 대체하는 VUI(Voice User Interface)의 필요성과 이를 가능하게 해줄 새로운 컴퓨터마크업 언어인 VoiceXML (Voice eXtensible Markup language)의 필요성이 대두되었다. 최근에 새로운 국제적 패러다임으로 모습을 나타낸 VoiceXML(Voice eXtensible Markup Language)은 XML에 기반을 두고 있으며 컴퓨터와 사람간의 대화를 정의할 수 있는 방안을 제시하였다. 웹을 기반으로 하는 대화형 음성 서비스를 구현에 있어 가장 효과적인 언어로서 주목 받고 있다[1][2][3].

본 연구에서는 음성 인식 소프트웨어와 VoiceXML 인터프리터를 기반으로 하여 VoiceXML로 상가검색 시스템을 구현하였다. 상가 검색을 구현하는 내부 처리 부분은 JSP와 JAVA Beans를 접목하여 구현하였으며 각각의 서비스 루틴을 컴포넌트화 하였다. 특히 사용자가 원하는 정보를 찾고자 할 때 걸리는 비용을 최소화하기 위해 사용자의 질의에 따라 다른 VoiceXML문서를 생성하게 하고, 사용자의 음성 입력 패턴을 정의하는 Grammar를 동적인 생성하게 함으로써 대량의 Grammar를 컴파일 할 때 걸리는 시간을 최소화하였다. 사용자가 자연어를 사용하였을 때 이를 처리하고 분석하여 사용자가 원하는 것을 정확하게

시스템에게 전달할 수 있게 하였다. 시스템이 사용자에게 적절한 프롬프트를 제공하게 하고, 각각의 경우에 대한 시나리오를 작성하여 보다 유연하고 자연적으로 시스템과 상호작용을 할 수 있도록 하였다[3][4].

또한 상가의 정보 및 Grammar를 관계형 데이터베이스로 설계하였고, 데이터베이스를 효율적으로 관리하기 위한 GUI 툴을 Python을 이용하여 구현 하였다.

2. 시스템 환경 및 구성

본 연구에서는 윈도우즈 NT4.0(SP 5.0)을 설치한 AMD 1GHz CPU와 768MB의 메모리 탑재한 시스템을 VoiceXMLGateway로 하였다. 음성 인식 소프트웨어로는 뉘앙스 음성인식 엔진(Nuance Speech Recognition System)7.0.4를 사용하였으며, VoiceXML 문서 인터프리터로 Nuance Voice Web Server1.3 (VWS)을 사용하였다

각종 스크립트와 오디오, Grammar를 저장 및 전송하는 웹 서버에는 JSP와 Java Beans를 운용할 수 있게 하기 위하여 Jakarta-Tomcat 3.2.3과 Java Standard Development Kit 1.3.1을 설치하였다. DBMS는 관계형 DBMS인 MS-SQL Server2000을 사용하였다.

시스템에 접속하는 사용자가 이용하는 장치로는 PSTN으로 접속하는 전화기를 이용 하는 것과 VoIP를 통한 H323 프로토콜을 사용하는 MS 윈도우 운영체제에 기본으로 설치되는 NetMeeting 3.0으로 접속하는 것으로 하였다.

그림 1은 본 연구에서 구성한 시스템의 구성도이다. 사용자가 VWS로 접속할 경우에, VWS는 먼저 사용자의 장치를 인식하고 웹 서버에 VoiceXML문서를 요청한다. 전송 받은 VoiceXML의 문서는 VWS에 의해 인터프리트된 후 정해진 다이얼로그를 수행 하게 된다. 사용자의 질의는 VWS를 거쳐 음성인식 엔진으로 전달된다. 음성인식 엔진은 Grammar를 컴파일 후 결과 값을 문자열로 VWS에 전달된다. 문자화된 사용자의 질의는 VWS가 VoiceXML 또는 JSP 를 요청하면서 파라미터로 전달된다. JSP는 전달 받은 파라미터 값을 DB에서 질의 내용의 키워드로 이용한다. 검색된 내용은 VoiceXML로 변환되어 다시 VWS에 전송 된다. 이와 같은 일련의 과정으로 사용자는 원하는 정보를 찾아가게 된다[5][6].

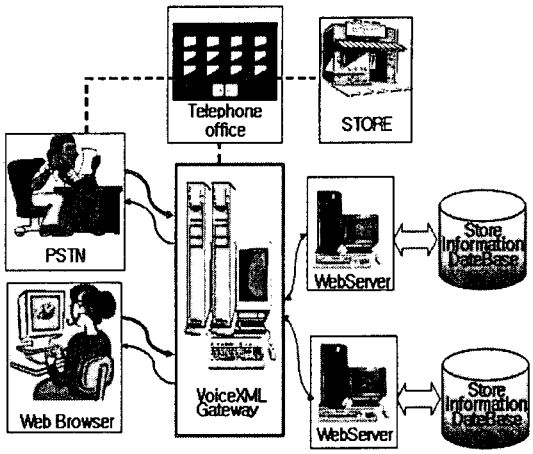

그림 1. Voice Store Locator 시스템 개요도

3. Call Flow Design

사용자의 질의에 따라 시스템의 응답이 결정되기 위해서는 자연어처리와 Call Flow의 계획이 필요하다. Call Flow는 사용자와 컴퓨터 간의 대화의 흐름을 미리 예상하고, 원하는 정보를 찾기 위한 최적의 경로를 나타낸 것이다. 사용자에게 어떤 프롬프트를 제공하느냐에 따라 사용자의 응답은 달라진다. 또한 음성 인식 엔진이 현재의 기술로는 사용자가 말하는 내용을 모두 완벽하게 인식할 수는 없다. 그러므로 엔진의 성능을 최대한 이용할 수 있게 하고, 사용자에게는 보다 빠르고 편리하게 원하는 정보를 찾기 위하여 자연어를 효과적으로 처리할 수 있도록 하였다.

자연어 처리 기능은 사용자가 말하는 문장에서 검색을 하기 위해서 필요한 키워드를 추출하는 것이다. 각 키워드를 종류별로 효과적으로 분류하고 Grammar의 크기를 적절하게 조절하게 하기 위해서는 Grammar의 내용이 동적일 필요가 있었다. 사용자가 말할 것으로 예상되는 단어나 문장이 Grammar에 들어가는데 이것의 크기가 너무 클 경우 컴파일 시간이 오래 걸리고 인식률의 저하를 초래하였다. 특히 지역을 '구' 단위로서 사용자가 찾고자 하는 상호를 검색할 경우 Grammar 컴파일 시간이 길어져 사용자의 대기 시간이 길어진다. 이 문제를 해결하기 위해서 대량의 Grammar의 경우에는 미리 컴파일을 하여 메모리상에 로드시키는 방법을 사용하였다[3].

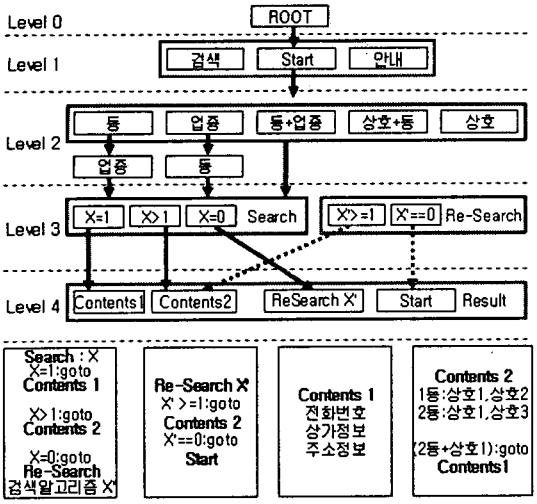

그림 2. Call Flow

그림 2는 각 상호 검색 시에 적용되는 Call Flow이다. 사용자가 말하는 형태를 레벨0을 제외한 레벨1에서 레벨4까지 총 4가지 단계로 구분하였다. 레벨1에서 사용자는 검색을 할 것 인지 아니면 도움말과 같은 안내를 받을 것인지를 선택하게 하였다. 레벨 2는 사용자가 동, 업종, 동과 업종, 상호와 동, 상호 중에서 한 가지를 말했을 경우이다. 각각의 경우에 따라서 레벨3의 다른 루틴으로 분기하여 각각 다른 정보를 사용자에게 제공하도록 하였다.

사용자가 원하는 상가를 검색할 수 있게 하기 위하여 상가를 찾는데 필요한 단어들을 추출한다. 입력 받은 단어를 키워드로서 검색을 하고 그 결과를 사용자에게 들려주게 하였다. 시스템과 사용자간에 질의와 응답에 의해 검색 범위를 좁히고, 최종적으로 사용자가 원하는 상가의 정보를 찾게 하였다. 이때 사용자가 찾고자 하는 상가가 없을 경우에는 시스템은 근접 지역 내에서 다시 검색을 수행하고 결과를 사용자에게 들려줌으로써 사용자가 실수를 보정 할 수 있도록 설계 하였다.

그림3은 상가 정보를 검색할 때 사용자가 원하는 상가가 없을 경우에 시스템이 사용자의 입력 중에서 힌트를 얻어 다른 지역이나 비슷한 이름을 가진 상가를 찾게 하는 것을 도식화 한 것이다. 사용자의 질의가 '동'과 '상호', '업종'에 대해 검색결과가 존재하지 않을 경우 시스템은 사용자의 질의를 실수로 판단하고, 사용자의 의도를 반영하여 재 검색 루틴을 수행한다.

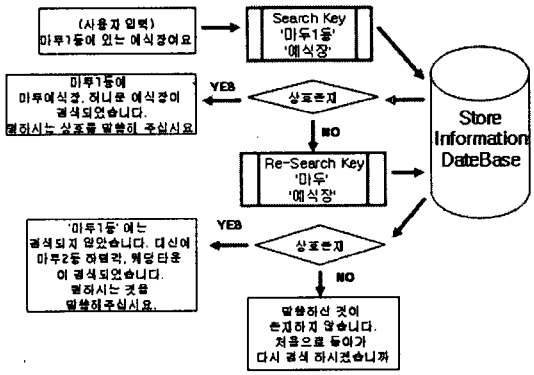

그림 3. 사용자 검색-재 검색 루틴

「'마두1동'에 있는 '예식장'을 찾아주세요」라는 사용자의 질의에 대하여 '마두1동'과 '예식장'이라는 검색 키 값을 가지고 수행을 한다. 사용자가 입력한 업종 또는 상호가 해당 범위(동)에 존재 할 경우 즉, '마두1동'에 '예식장' 이 존재 할 경우 검색은 한번만 수행이 된다.

결과가 존재 하지 않을 경우 주위 동을 검색하는 루틴이 수행된다. 예를 들어 사용자가 '마두1동에 있는 예식장 찾아주세요'라고 질의를 요청했을 때, 만약 StroeInfo DB에 '마두1동'에 속한 예식 업종이 존재하지 않을 경우에는 시스템은 마두동, 마두2동 등 '마두'란 이름의 '동'에 등록 되어있는 예식 업종을 검색하고 해당 '동'과 '상호'를 들려준다. 접근도가 낮고, 인구수용도가 높은 업종에 포함된 상호는 소수일 경우로 판단하고 검색 결과 전부를 사용자에게 들려준다.

검색과 재 검색 루틴을 수행함으로서 사용자의 업종 또는 상호의 지리적 정보가 명확하지 않는 검색키를 입력 하였을 경우에도 시스템은 이를 해결해 줄 수 있었다.

4. 동적 VoiceXML 및 Grammar 생성

VoiceXML은 사용자와 컴퓨터간의 대화를 정의하고 표현하는 것이다. 각 다이얼로그를 모두 VoiceXML로 작성하여 표현할 수도 있으나 다양한 다이얼로그와 음성정보를 표현 하는데 있어서 한계가 있었다. 또한 VoiceXML은 선언적 마크업 언어라

는 특성상 시스템의 내부처리까지 할 수는 없고, DBMS 와의 통신이나 입력 받은 정보의 처리가 거의 불가능하다. 이를 위한 해결 방법으로 클라이언트 시스템의 정보의 이용 및 알고리즘과 연산처리 그리고 DataBase와의 연동의 구현을 위해 ECMAScript를 이용할 수 있다. 서버와 클라이언트의 응답속도를 최우선으로 고려한다면 ECMAScript를 사용하는 것이 더욱 유리하겠으나 이것은 표준 VoiceXML을 사용할 수가 없게 된다. ECMAScript는 VoiceXML의 인터페이스 부분과 처리부가 합쳐지는 결과를 초래하여 개발과 유지, 보수 시에 문제점이 발생하게 된다. 또한 ECMAScript를 지원하는 VoiceXML 인터프리터와 Database 드라이버 등이 별도로 구현되어야 한다는 문제점이 발생할 수 있다

본 연구에서는 JSP와 자바 빈즈를 이용하여 위의 단점을 개선하고 다양한 VoiceXML과 Grammar를 표현할 수 있도록 하였다. JSP는 VoiceXML과 Grammar의 표현을 위해 사용하였고 Java Beans는 구분된 각 시스템의 내부처리를 컴포넌트구조로 구현하였는데 본 연구에서 사용한 JSP와 자바 빈즈는 자바의 플랫폼 독립성과 안정성, 객체지향적 특성을 이용할 수 있어 타 시스템과 호환성 및 재사용성이 우수하였다[3][8].

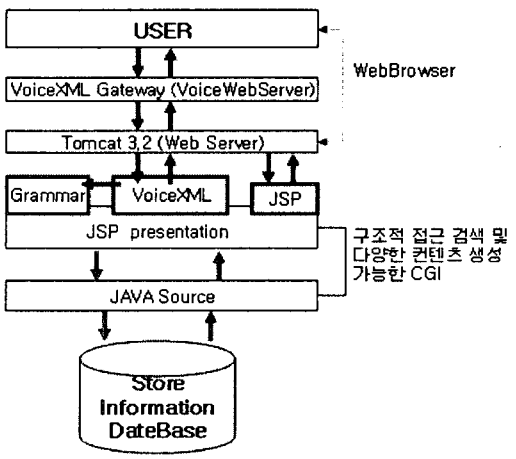

그림 4. 동적 VoiceXML과 Grammar 생성 로직

그림4는 JSP와 자바 빈즈로 데이터베이스에서 상가정보를 검색하여 이를 VoiceXML과 Grammar로 표현하는 절차를 각 계층별로 분류한 것이다. 자바 클래스로 만든 자바 빈즈는 데이터베이스에서 상가 정보를 검색하거나 음성 정보를 제공하는데 필요한 내부 처리를 위한 컴포넌트이다. JSP는 이 자바 빈즈를 내부 페이지에 포함하게 되면 JSP/Servlet 컨테이너인 톰캣 서버에서 컴파일되어 VoiceXML이나 Grammar의 표현 형식으로 변환되어 VoiceXML 인터프리터로 전송된다.

```
http://localhost:8080/MAS/dialogs/D_Name/USE_SEARCH.jsp?bc3_id=0200100002&dong=마두1동
<?xml version="1.0" encoding="EUC-KR" ?>
<!--                                                                    -->
<!--                                                                    -->
<!--                                                                    -->
<!DOCTYPE vxml (View Source for full doctype...)>
<!--                                                                    -->
- <vxml version="1.0" xml:lang="ko-KR">
  - <form id="start">
      <grammar src="US_grammar.jsp?bc3_id=0200100002&dong=마두1동#NAME" />
    - <block>
        <audio src="../../prompts/DONG/마두1동.wav" />
      - <prompt bargein="true">
          <audio src="../../prompts/ETC/3.1.2.wav" />
        </prompt>
      - <prompt bargein="true">
          <audio src="../../prompts/ST_NAME/대전갈비.wav" />
          <audio src="../../prompts/ST_NAME/녹산가든.wav" />
          <audio src="../../prompts/ST_NAME/계미식당.wav" />
          <audio src="../../prompts/ST_NAME/기사님식당.wav" />
          <audio src="../../prompts/ETC/3.1.3.wav" />
          <audio src="../../prompts/ETC/3.2.0.wav" />
        </prompt>
        <assign name="dong" expr="마두1동" />
      </block>
```

그림 5. 마두1동의 한식집들을 사용자에게 들려주고 <grammar>에 의하여 선택하게 하는 VoiceXML의 예

그림5는 사용자가 마두1동에 있는 한식집의 목록을 요청했을 때 생성되는 VoiceXML문서를 인터넷 익스플로어 6.0으로 출력한 것이다. 이 VoiceXML문서는 실제로 마두1동에 한식집이 어떤 것이 있고 이것 중에서 원하는 한식집을 선택하는 다이얼로그를 수행하도록 VoiceXML인터프리터에 의해 처리된다. 일반 웹 브라우저에서 출력결과로 보인 것은 실제 VoiceXML 인터프리터가 웹으로 요청한 JSP가 어떤 형식의 VoiceXML로 전달 받는지를 보여주기 위해서이다. <grammar/>는 웹서버에서 Grammar파일을 요청하는 태그이다. 이것을 직접 VoiceXML문서에서 작성할 수도 있으나 Grammar의 크기가 크고 다양한 표현을 위해서 그림6과 같이 JSP로 작성이 필요하다.

그림6의 JSP는 vxmlkorea.info.USE_SEARCH_DB라는 이름으로 구현한 자바 빈즈 클래스를 호출 후에 dbc라고 명명한 새로운 객체를 생성한다. 자바 빈즈 내에는 그림7과 같은 GSL(Grammar Specification Language) 형태의 Grammr를 생성하는 메서드들이 있다. 이 메서드들에 각각의 파라미터 값들이 들어감으로 인해서 다양한 형태의 Grammar가 생성되게 된다. 각 자바 빈즈들을 컴포넌트 구조로 구현하였고 클래스파일로 컴파일을 해둔 상태이므로 재 컴파일시에 소요되는 시간을 줄일 수 있다.

```
<%vxmlkorea.info.USE_SEARCH_DB dbc;
dbc = new vxmlkorea.info.USE_SEARCH_DB(out);%>

<%BusinessID=request.getParameter("BusinessID");%>
<%Dong=request.getParameter("Dong");%>

<%dbc.DB_BusinessID = BusinessID;%>
<%dbc.DB_Dong=Dong;%>

<%dbc.USD();%>
```

그림 6. JSP에 사용자가 입력한 동과 업종을 검색 하는 US_Grammar.jsp 파일

그림7의 GSL은 음성 인식 엔진에서 입력되고 사용자의 음성 입력을 대기하는 단어를 GSL형식에 맞게 표현한 것이다. Grammar는 상가의 이름과 동을 분류하도록 Sub-Gramamr인 ST_NAME과 DONG으로 나누었으며 인터넷 익스플로서 6.0에서 텍스트 형식으로 확인을 한 것이다.

```
NAME
[
{?{ST_NAME:s} ?[MiddleSyntax] ?{DONG:d} ?[LastSyntax]} {<gst_name $s><dong $d>}
{?{DONG:d} ?[MiddleSyntax] ?{ST_NAME:s} ?[LastSyntax]} {<gst_name $s><dong $d>}
]
ST_NAME[
{대진갈비} {<gst_name "대진갈비">}
{녹산가든} {<gst_name "녹산가든">}
{계미식당} {<gst_name "계미식당">}
{기사님식당} {<gst_name "기사님식당">}
]
DONG
[
{마두일동} {<dong "마두1동">}
]
```

그림 7. US_grammar 에 의해 생성된 GSL형식의 Grammar

Grammar는 사용자의 음성을 입력받고 그에 해당하는 값을 ViceXML에 리턴해주는 역할을 하고 있다. 따라서 VoiceXML에서 Grammar의 역할은 시스템과 사용자와의 입력 인터페이스를 수행한다. 시스템은 사용자에게 그림6의 단어들 중에서 선택을 하도록 정보를 제공한다. 자연어 처리를 위해 단어 사이와 문장의 끝에 각각의 MiddleSyntax와 LastSyntax를 코딩하여 사용자가 어떠한 조사와 끝마침을 하더라도 이에 상관하지 않고 '업종', '동'의 키워드를 추출할 수 가 있게 하였다. '?'는 각 키워드가 한번 입력되거나 입력되지 않음을 의미한다. 이와 같은 정의를 함으로써 각각의 분류된 키워드 중에서 사용자가 말한 것만 추출할 수 있는 것이다.[7]

5. 상가 정보 및 Grammar 데이터베이스 관리 툴의 구현

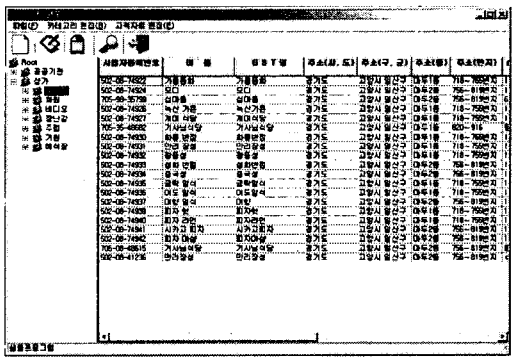

그림 8. DB 관리 툴 메인 화면

본 연구에서 구현한 상가 및 Grammar 데이터베이스 관리 툴은 상가의 종류가 많아지고 사용자가 입력할 수 있는 다양한 Grammar를 등록시에 편리성을 두기 위해 구현하였다. GUI Addon으로는 wxPython을 사용하게 되었으며, 컴파일러는 Py2exe를 선택하여 MS-Windows의 GUI 인터페이스가 될 수 있도록 하였다. 데이터베이스의 접근 처리를 위해 MS-SQL을 지원하고 동시에 Python DB v2.0을 지원하는 MSSQL module을 선택 사용하였다.

그림8의 좌측의 트리구조 메뉴는 상가의 효율적인 분류를 위해 구현 한 것이다. 이러한 카테고리는 한국 YellowPage의 규격을 채택하였다. 가장 상위부터 [상가 - 업종 - 세부업종 - 상호명] 으로 트리가 나누어져 있다.

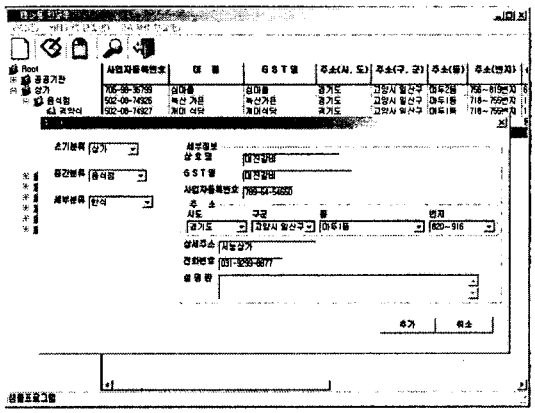

그림 9. 상호에 해당하는 정보 및 Grammar 입력 폼

그림9의 입력 폼 에는 필수 기재 사항이 있으며 GST(Grammar Store Table)명을 구현하였다. 초기 분류, 중간 분류 세부 분류의 카테고리를 두었다. 세부 분류의 경우에 사용자로부터 입력받을 수 있는 질의는 같은 업종이라도 동의어가 존재한다. 예를 들어 중국집, 중화요리, 자장면집 등이 이에 해당하며 이것을 Grammar에 추가하여 인식률을 높였다. VWS에서 인식하는 Grammar는 매우 적법해야한다. 즉, Nuance 엔진의 경우 한글을 인식하기 위해서는 한글 이외의 모든 문자는 한글 발음으로 DB에 입력해야 한다. 예를 들어 'OB1 라운지' 라는 상호명은 GST명에 '오비원 라운지' 또는 '오비일 라운지' 라고 입력을 해야 한다. 또한 사업자 등록번호의 검증을 할 수 있게 했다.

상호명은 DB에 입력 된 형과 데이터를 사용하는 것이 아니다. 사용자의 질의가 있은 후에 처리과정시 JAVA Beans내에서 한글변환과 공백, 특수문자를 제거하는 루틴을 수행할 수 있다. 그러나 루틴이 수행되어 출력되기까지의 시간은 사용자의 대기시간으로 돌아가게 된다. 즉, 사용자의 시간은 검색시간과 한글 변환출력 시간이 추가 된다. 그러므로 미리 GST 테이블을 만들고 Grammar에 적법한 한글 입력하여 사용자가의 검색 후 데이터를 바로 청취 할 수 있게 하였다.

6. 결 론

음성을 중심으로 음성의 인식과 합성을 이용한 응용 시스템은 휴먼 인터페이스로 각광을 받고 있다. 본 논문은 음성인식엔진을 이용하여 음성 상가 검색 시스템을 CGI언어인 JAVA와 VoiceXML을 기반으로 구현하여 사용자와 시스템의 상호 인터렉티브 통신을 할 수 있게 하였다. 또한 자연어 처리와 사용자 실수의 보정을 하여 재 검색을 할 수 있도록 구현하였다. 특히 JAVA Beans를 사용함으로서 코드의 재사용과 클래스의 재배포성을 유지하여 효율적인 개발을 할 수 있었으며, VoiceXML의 기본 목표에 충실하게 음성시나리오 작성과 데이터 처리부분을 나누어 개발과 관리측면을 용이 하게 하였다. 관리의 효율성과 VoiceXML의 Grammar규칙을 고려하여 음성 상가 검색시스템의 한 부분으로 상가 정보 데이터관리 프로그램을 구현하였다.

음성 인식 엔진의 발전은 더욱더 다양한 VUI 서비스를 개발하게 할 것이며 CGI 언어들과의 상호 연동을 하여 처리속도와 개발의 편의성을 증진 할 것으로 기대 된다.

[참고문헌]

[1] Voice extensible Markup Language VoiceXML, Version 1.0, VoiceXML Forum, Mar. 2000, <http://www.voicexml.org>.

[2] Nuance Developer Networks, <http://extranet.nuance.com>.

[3] 장준식, 김민석, 윤재석 "VoiceXML을 이용한 음성 인식 시스템에서의 ASP모듈연구" 한국 해양 정보통신학회 논문지, Vol.5, No.2, pp.609 set2001.

[4] JavaServer Pages(TM) Technology, <http://java.sun.com/products/jsp/index.html>

[5] Nuance Voicewebserber Guide, <http://www.nuance.com/pdf/voicewebserver_tech.pdf>

[6]Voxeo Developer Community, <http://community.voxeo.com>

[7] Nuance Grammar Developer's Guide, <http://extranet.nuance.com>.

[8] Voice Application Development with VoiceXML by Rick Beasley, Kenneth Michael Farley, Johon O'Reilly Leon Squire, Kenneth Farley, Indianapolis, Ind : SAMS , 2001.