

VoiceXML을 이용한 대화형 음성 인식 증권 거래 시스템 구현

조 창 수* · 신 정 훈** · 홍 광 석***

요 약

본 논문에서는 음성 서비스 구현 시 개발시간을 단축시킬 수 있으며, 개발된 서비스간의 콘텐츠를 보다 자유롭게 공유할 수 있는 VoiceXML을 이용하여 응용서비스를 구현하였다. 현재까지의 음성 서비스 개발은 시스템 의존적인 API의 활용 및 C/C++ 등의 프로그래밍 언어나 전용 개발 도구를 사용하여 이루어져 왔다. 이로 인하여 각 응용서비스간 자원의 재활용이 어려운 실정이었다. 응용 서비스의 내용이 변경되어지거나 시스템이 바뀌게 되면 다시 프로그램을 작성하거나, 적절한 API로의 수정이 필요하였다. 이러한 문제점을 해결하기 위하여 최근에는 VoiceXML을 이용한 서비스 개발이 증가하고 있는 실정이다. 그러나 VoiceXML을 이용한 응용 서비스 구현시 실제 환경에서 발생하는 문제점 및 해결 방안에 대해서는 정확한 파악이 미비한 상황이다. 본 논문에서는 이러한 문제점들을 규명, 해결하기 위하여 VoiceXML에 기반을 둔 음성 인식 증권 거래 시스템을 구현하였으며, 서비스 구현 시 발생하는 문제점과 해결방안을 제시하였다. 또한, 구현 되어진 시스템을 토대로 응용서비스 및 시스템의 장단점을 분석한다.

An Implementation of Interactive Voice Recognition Stock Trading System Using VoiceXML

Chang-Su Cho* · Jeong-Hoon Shin** · Kwang-Seok Hong***

ABSTRACT

In this paper, we implemented practical application service using VoiceXML. Developers can utilize the advantages of using VoiceXML such as reducing development time and sharing contents between applications. Up to now, speech related services were developed using APIs and programming languages such as C/C++ or exclusive developing tools, which methods depend on system architectures. For this reasons, reuse of contents and resources was very difficult. If developers want to change scenarios of the application services or change platforms, they have to edit and recompile their program sources. To solve these problems, nowadays, companies develop their applications using VoiceXML. But, there's poor grip of actual problems can be occurred when they use VoiceXML. To overcome these problems, we implemented stock trading system using VoiceXML. We found out problems which occurred during developing services. We proposed solutions to these problems. And, we analyzed strong points and weak points of applications using suggested system.

키워드 : 음성 인식(Voice Recognition), VXML, VoiceXML

1. 서 론

VoiceXML은 XML을 이용한 특화된 마크업 언어의 일부로서, XML(eXtensible Markup Language)은 W3C(World Wide Web Consortium)가 HTML 한계를 극복하기 위한 대안으로 만든 마크업 언어이다. AT&T, IBM, Lucent Technologies, Motorola 등이 설립한 VoiceXML 포럼에서는 1999년 8월 음성 응용 프로그램 개발을 위해 VoiceXML(Voice eXtensible Markup Language) 0.9 버전을 발표하였고, 2000년

3월 이를 보완한 1.0 버전을 정식으로 제안하였다. W3C에서는 이 제안을 받아들여 2000년 5월 웹의 대화형 마크업 언어로서 VoiceXML을 표준으로 공인하였다. 현재 1.0 버전의 문법 부분을 보완한 VoiceXML 2.0버전이 나와 있다[1].

VoiceXML을 이용한 개발의 이점은 음성 관련 응용 서비스를 개발할 때 웹 개발 기술과 웹 콘텐츠 전송 기술을 최대한 활용하고, 음성 응용 프로그램 개발자들이 힘들어했던 저수준(low-level) 프로그래밍과 자원관리에 소요되는 노력을 덜어줄 수 있다는 데 있다. 전화를 인터페이스로 하는 음성 정보 서비스의 콘텐츠 및 응용 프로그램 개발 시 기존 개발 환경은 C나 Assembly와 같은 저수준 프로그래밍 언어(low level programming language)와 시스템 의존적인

* 준 회 원 : 성균관대학교 대학원 정보통신공학부

** 준 회 원 : 인덕대학교 정보통신전공 교수

*** 종신회원 : 성균관대학교 정보통신공학부 교수

논문접수 : 2004년 3월 22일, 심사완료 : 2004년 7월 13일

API(Application Program Interface)를 사용하였다. 이로 인하여 응용 프로그램의 개발 시간 증가와 각 서비스 시스템 간의 호환성 관련 문제점이 존재 하였으며, 시나리오 변경 요구시 전체적인 프로그램의 재 수행을 해야하는 문제점이 발생하였다. 또한 음성 정보 서비스 개발자에게는 음성 인식 및 합성과 같은 전문적인 지식이 요구 되었으며, 시각에 의존적인 웹을 벗어나 음성 및 시각을 모두 활용할 수 있는 새로운 틀을 요구하였다[2].

이런 문제점 해결을 위하여 VoiceXML이 개발되었다. VoiceXML을 이용한 음성 응용 서비스 개발은 서비스 내용이 바뀔 때마다 콜 플로우 시나리오를 새로이 작업해야 하는 것과 같은 문제점들을 사라지게 되었다. 예를 들어 필요에 따라 홈페이지의 문서를 수정 또는 삽입하는 것과 같은 간단한 방법처럼 시나리오를 재작성할 수 있게 되었으며, 개발시간의 단축 및 수정, 갱신, 자원의 재활용이 편리하게 되었다. XML기반 자료의 경우 XSLT(eXtensible Stylesheet Language for Transformation)를 이용해 VoiceXML로 변환할 수 있다. 즉, 원본 XML자료를 XSLT를 이용해 원하는 자료포맷(WAP, HTML, VXML)으로 변환할 수 있기 때문에 음성 서비스 구축시 개발 및 유지보수가 용이해 졌으며, VXML표준을 사용하기 때문에 어떠한 플랫폼에서도 호환이 가능하게 되었다. 단일한 소스파일의 관리를 통해 웹, 유/무선전화, PDA 등 다양한 사용자 접속 환경을 만족 시킬 수 있게 되었다.

또한, VoiceXML은 클라이언트 측에서 연산기능 및 함수를 구현할 수 있는 ECMAScript를 지원하며 ASR(Automatic Speech Recognition), TTS(Text To Speech) 등을 지원하기 때문에 기존 시나리오보다 다양하고 정교한 시나리오를 구성할 수 있다. 웹에서 이미 구축된 인프라를 토대로 시각으로만 서비스를 제공하던 많은 업체들이 적은 노력으로 그들의 서비스를 새로운 방식(Voice/Speech)으로 서비스할 수 있게 되었으며 인터넷의 발전과 같이해 온 웹 관련 기술들을 자신의 서비스에 대부분 사용할 수 있게 되었다. 웹 사이트에서 개인별 서비스가 가능한 것처럼 VoiceXML을 이용한 전화망 서비스도 개별 서비스가 가능하다. 이러한 이유로 VoiceXML은 웹 사이트뿐 아니라, 기존 CTI(Computer Telephony Integration)분야의 IVR(Interactive Voice Response) 한계를 극복하는 유력한 대안이라고 볼 수 있으며 무선 이동통신 단말기의 핸즈프리 어플리케이션 분야에도 응용할 수 있다. 이러한 장점으로 인하여 여러 회사에서 응용서비스를 개발 시 VoiceXML을 활용하고 있는 상황이며, 대표적인 예로는 Tellme(<http://www.tellme.com>)를 들 수가 있다. 이 업체는 전화로 인터넷상의 정보 획득 서비스를 제공하고 음성 자동 다이얼링(VAD, Voice Activated Dialing) 기술을 이용해 음성으로 주소록을 검색하고

주소록의 전화번호를 이용해 전화를 걸어주는 서비스를 제공한다. 이외에 VoiceGenie(<http://www.voicegenie.com>), HeyAnita (<http://www.heyanita.com>), BeVocal(<http://www.be-vocal.com>)등이 있다. 또한 전화망/음성인식 서비스 업체뿐만 아니라 웹 서버 시스템 업체들도 VoiceXML에 대한 관심을 높여가고 있는 실정이다. IBM은 자사의 Websphere 서버와 VoiceXML로 서비스를 제공하고 있으며 국내에는 KT의 1588-0114의 음성 다이얼 서비스, 와이더텐닷컴(<http://vxml.widerthan.com>)의 SK 텔레콤 n.TOP Voice Service가 운용 중에 있으며 미디어포드(<http://www.mediaford.co.kr>)의 전화 결제 시스템 등 여러 서비스를 제공하고 있고, 텔미텔미(<http://www.telmetellme.com>)는 국내 사이버 개인비서로 웹과 전화를 이용한 개인정보 관리 서비스를 제공하고 있다[3]. 이외에도 에이블컴(<http://www.ablecom.co.kr>), 브레인투엔티원(<http://www.brain21.com>)등이 있다.

이상에서 서술한 바와 같이 VoiceXML을 사용하여 응용 서비스를 구현하면, 많은 이점들이 존재하기에 이를 활용한 서비스 개발이 증가하고 있는 실정이다. 그러나, 이러한 이점들에 반하여 VoiceXML을 이용한 서비스 구현 시 발생할 수 있는 문제점과 해결방안에 대한 정보 공유가 부족한 실정이다. 본 논문에서는 웹 서버와 데이터베이스를 구축하고 VoiceXML과 ASP(Active Server Pages)를 이용한 음성인식 증권 거래 서비스를 구현하였다. 응용 서비스 구현시 발생 가능한 문제점을 파악하고 해결 방안을 제시하였으며 음성 인식 증권거래 시스템 운용에 따른 결과를 제시함으로써 개발자들에게 실질적인 도움을 주고자한다.

본 논문에서 구현한 "VoiceXML을 이용한 대화형 음성 인식 증권거래 시스템"의 경우 운용자 편의를 위하여 안내 음성 수정 또는 갱신할 경우 해당 VoiceXML 파일을 직접 수정하지 않고 웹 페이지를 통하여 실시간으로 정보 입력 및 수정 기능을 제공하였다. 본 논문의 구성은, 2장에서 VoiceXML의 구성 및 VoiceXML을 이용한 기존 대화형 증권 서비스의 구성환경에 대하여 서술하였으며, 3장에서는 본 논문에서 구현한 대화형 증권 서비스의 구성환경 및 서비스 흐름, 실질적인 운용결과 및 문제점 해결방안에 대하여 제시하였으며, 4장에서 결론 및 향후 연구 방향에 대해 기술하였다.

2. VoiceXML

일반적으로 우리는 웹상의 정보를 얻기 위해서 컴퓨터의 웹 브라우저를 통해 원격지에 있는 웹 서버로 정보를 요청하면 웹 서버는 URL의 내용을 보고 사용자가 요청한 파일을 사용자에게 전송하여 모니터로 보여준다. 반면 전화를 이용하여 웹의 정보를 얻기 위해서는 VoiceXML 게이트웨

이 컴퓨터의 웹 브라우저의 역할을 대신 하게 된다[4]. 좀 더 구체적으로 설명하면 VoiceXML 게이트웨이는 웹 서버에 URL을 전송하고 웹 서버가 보낸 VoiceXML 문서를 분석하여 사용자에게 음성이나 오디오 신호를 전화망을 통해 사용자에게 전달한다.

(그림 1) 대화용 음성 인터페이스 구성도

(그림 1)은 사용자가 전화를 이용하여 웹서버에 있는 문서의 정보를 음성을 통해 얻는 것을 나타낸 그림이다. 전화기는 VoiceXML 문서를 분석하는 기능이 없기 때문에 전화기와 웹 서버 사이에 VoiceXML 게이트웨이라 불리는 장치가 필요하다.

2.1 VoiceXML 게이트웨이

VoiceXML을 이용하여 음성 응용 서비스 제공 시 VoiceXML 게이트웨이는 필수 요소이다. VoiceXML 게이트웨이에는 전화 접속 기능과 VoiceXML 문서를 해석하는 VoiceXML 해석 기능이 있어야 하며, 사용자의 음성 입력을 받기 위해서 음성 인식이 필요하고 텍스트로 기술되어 있는 정보를 사용자에게 들려주기 위해 음성 합성기가 필요하다.

VoiceXML 게이트웨이는 전화망과 인터넷 망을 연결해 주는 역할을 하고 웹브라우저처럼 웹 서버에 URL을 전송하고 웹 서버가 보낸 VoiceXML 문서를 분석하고 렌더링하는 기능을 수행한다[5, 6].

또한, 전화망과의 접속을 위하여 하드웨어가 필요하며 이 하드웨어는 전화선 인터페이스를 제공하는 몇 개의 포트를 가지고 있다. 제조사 및 제품별로 제공되는 포트수가 다른데 참고로 Dialogic D/120JCT-LS 같은 경우 1개의 Board에 12개의 포트를 가지고 있으며 한 시스템에 120포트까지 확장할 수 있도록 10개까지의 Dialogic Board를 설치할 수 있다.

2.1.1 VoiceXML 해석기

VoiceXML 해석기는 입력된 VoiceXML 문서를 DOM(Document Object Model) 트리로 구성하여 문서의 엘리먼트간의 계층 구조와 구문이 올바른지 아닌지를 검사하는 검증기, DOM 트리 노드를 순회하면서 메뉴 해석 알고리즘과 폼 해석 알고리즘을 적용하여 해석한 후 음성 인식기와 음성 합성기에게 입력력을 명령하는 의미 실행기로 구성된다. DOM은 XML 문서를 처리할 때 공통적으로 사용할 수 있는 중요한 표준이다[6, 8].

VoiceXML 해석기에서 검증기는 구문을 검증한 후 의미 실행기에서 다이얼로그, 문법, 이벤트, 오디오 출력, 오디오 입력, 콜 제어, 흐름 제어와 관련된 47개의 엘리먼트 기능에 따라 VoiceXML 문서에 작성된 음성 서비스 시나리오를 해석하고 음성 플랫폼의 각종 음성 입력력 기능들을 호출하면서 사용자와의 대화 흐름을 제어한다. 대화 내용에 따라 다른 문서로의 전환, 각종 정보의 업로드 및 다운로드를 총괄 지휘한다. (그림 2)는 VoiceXML 해석기 구성도를 나타낸다.

(그림 2) VoiceXML 해석기 구성도

2.1.2 음성 인식기

음성은 가장 편하고 자연스러운 의사 전달 수단 중 한가지이며 사람과 기계간의 인터페이스로써 활용성이 증대되어지고 있다. 음성 인식이란 음성에 포함된 음향학적 정보로부터 언어적 정보를 추출하여 이를 기계가 인지하고 반응하게 만드는 일련의 기술이다. 음성은 입력 및 전달 과정에서 고가의 장치를 필요로 하지 않는다는 장점을 가지고 있으며 최근 컴퓨터의 급속한 발전으로 음성을 매개체로 하는 사람과 기계간의 인터페이스를 음성으로 채택한 많은 제품들이 상용화 되어지고 있다.

VoiceXML 게이트웨이에서 음성 인식기는 VoiceXML 해석기가 VoiceXML 문서를 해석한 다음 음성입력을 받아야 할 경우 음성 인식을 위해 음성 인식기가 가동되어지며, 인식에 필요한 자원을 읽는다. 사용자의 음성이 입력으로 들어오면, 입력되어진 사용자의 음성과 인식후보와 비교 후, 인식 결과 값을 VoiceXML 해석기로 전달한다.

2.1.3 음성 합성기

음성 합성 기술의 목적은 다양한 형태의 정보를 실제 인간의 음성과 유사하도록 명료성과 자연성을 가진 음성으로 합성하여 정보를 전달하는데 있다.

음성 합성기는 휴먼-컴퓨터 인터페이스의 일환으로 인간의 발성 원리를 컴퓨터에 실현하여 컴퓨터가 텍스트 문장을 인간의 음성으로 합성하여 출력시키는 방식으로 비교적 오래 전부터 연구가 진행되어 왔다. 현재 많은 국가에서 자국의 언어로 된 음성 합성에 대해서 연구하고 있으며 상용 제품도 많이 출시되었다.

합성원리는 VoiceXML 해석기가 VoiceXML문서를 해석한 다음 음성 출력을 해야 할 경우 문장을 음성 합성기에 입력하여 사용자에게 음성을 전달한다. (그림 3)은 VoiceXML의 시스템 구성도를 나타낸 그림이다.

(그림 3) VoiceXML의 시스템 구성도

2.2 VoiceXML 엘리먼트

문서 타입을 기술할 수 있고 유연성을 가지고 있는 XML에 비하여 VoiceXML은 47개의 정해진 엘리먼트와 그 엘리먼트의 정해진 속성값, 그리고 엘리먼트 간에 계층구조가 있다는 제약 조건을 가진다.

VoiceXML의 엘리먼트간의 계층 구조는 명확한 상위 엘리먼트와 하위 엘리먼트 개념으로 이루어져 있는데, 예를 들어 (그림 4)의 문서가 입력되면 입력된 문서의 유효성 여부를 체크 후, VoiceXML의 계층 구조가 맞는지 점검하고 지정된 속성값을 가지는지 검사한다. 즉 <vxml> 엘리먼트의 하위 엘리먼트로써 <link>, <form>이 올 수 있는지 점검하고 <form> 엘리먼트의 하위 엘리먼트로써 <field>가 올 수 있는지 점검을 한다. 만약 어떤 메시지를 <prompt> 엘리먼트로 출력하려면 적어도 2개의 상위 엘리먼트(<vxml>, <menu>)가 존재하여야 한다.

또한 XML과 달리 VoiceXML의 엘리먼트중에는 속성값을 가지는 엘리먼트들이 있는데 이 속성값도 계층 구조와

함께 조사하게 된다. 예를 들어 <prompt>는 합성된 음성을 출력하는데 쓰이는 엘리먼트로서 4개의 사용가능한 속성값을 가지고 있다. 이 중 bargein은 안내멘트 중간에 사용자의 입력을 허용할 것인지 말것인지를 지정하는 것으로써 bargein = "true"이면 허용하는 것이며 bargein = "false"이면 허용하지 않게 된다. 이 값에 대한 기본값은 true로 설정되어 있다. timeout 속성 값은 안내멘트가 실행되는 도중에 사용자의 대답을 기다리는 시간을 지정하는 값으로써 이 시간동안 사용자의 응답이 없을 경우 noinput 이벤트가 발생한다. 기본값은 VoiceXML 게이트웨이다 상이하게 설정되어져 있다. 사용단위는 1000분의 1초인 "ms", 초는 "s", 분은 "m", 시간은 "h"와 같은 문자를 사용한다. 사용 시 원하는 정수 다음에 단위를 나타내는 문자를 함께 사용하게 되어있다.

```

<? xml version = "1.0" encoding = "EUC-KR" ?>
< vxml version = "1.0" >
  < meta name = "author" content = "Chang Su Cho" />
  < link >
    < dtmf > * </ dtmf >
  </ link >
  < form id = "stock" >
    < field name = "menu" >
      < prompt bargein = "true" timeout = "5s" >
        무엇을 도와드릴까요? 시세안내, 주문거래, 체결안내,
        계좌정보, 종료가 있습니다.
      </ prompt >
      < grammar src = "ment.gram" > </ grammar >
      < noinput > 죄송합니다. 입력데이터가 없습니다.
      < reprompt />
      </ noinput >
      < nomatch > 죄송합니다. 다시 한번 말씀해 주세요.
      </ nomatch >
      < filled >
        < if cond = "menu == '시세안내'" >
          < goto next = "/quotation/quotation.vxml" />
        < elseif cond = "menu == '주문거래'" />
          < goto next = "/order/order.vxml" />
        < elseif cond = "menu == '체결안내'" />
          < goto next = "/conclusion/conclusion.vxml" />
        < elseif cond = "menu == '계좌정보'" />
          < goto next = "/account/account.vxml" />
        < elseif cond = "menu == '처음'" />
          < goto next = "/menu/ment.vxml" />
        < elseif cond = "menu == '종료'" />
          < goto next = "/menu/stock/endment.vxml" />
        </ if >
      </ filled >
    </ field >
  </ form >
</ vxml >
    
```

(그림 4) VoiceXML 문서의 예

VoiceXML 엘리먼트은 <표 1>과 같이 크게 10개로 분류할 수 있으며, 47개의 각각의 엘리먼트를 기능별로 분류하면 다음과 같다.

〈표 1〉 VoiceXML 엘리먼트의 기능별 분류

분 류	요 소
문 서	vxml, meta
다이얼로그	form, menu, choice
프롬프트	prompt, enumerate, reprompt
필드	field, option, var, initial, block, assign, clear, value
이벤트	catch, error, help, link, noinput, nomatch, throw
출 령	audio, break, div, emp, pros, sayas
입 령	dtmf, grammar, record
호제어	disconnect, transfer
흐름제어	if, elseif, else, exit, filled, goto, param, return, subdialog, submit
객 체	object, property, script

2.3 기존 대화형 증권 정보 서비스 구성환경

사람과 컴퓨터 간의 다리 역할을 하는 IVR 시스템은 이미 10여년 전에 보급되기 시작하여 현재는 주식거래, 여행예약 및 은행 계정 관리 등을 지원하는데 이용되고 있다. 예전에는 DTMF(Dual-Tone Multifrequency)로 사용자 인터페이스를 채용하고 있었지만 현재는 음성 인식 기술의 발전으로 사용자들이 DTMF 입력과 자신의 음성을 이용할 수 있도록 인터페이스가 제공되어지고 있는 실정이다. 대부분 음성 서비스는 CTI 서버에 응용 서비스가 하드 코딩되어져 있다. 이러한 구조로 인해 서비스를 제공하는 사업자가 직접 서비스의 내용을 수정하기가 어려웠고 다른 서비스와 연동 또한 쉽지 않은 실정이다. (그림 5)는 기존의 대화형 음성 서비스 시스템 구조를 나타낸 그림이다.

(그림 5) 기존 대화형 음성 서비스 시스템 구조

3. VoiceXML을 이용한 증권 거래 시스템 구현

최근의 추세를 분석해 보면 전화를 경유한 음성으로 개인 일정 관리, 음성 다이얼 등의 음성 서비스를 제공하는 인터넷 사이트가 증가 하고 있는 실정이다. VXML포럼에서는 음성사용이 전자상거래 장착의 마지막 숙제라고 파악하고 있을 정도로 이 부분에 대한 관심과 기대가 크다.

본 논문에서는 VoiceXML을 이용하여 전자상거래의 일부분인 음성 인식 증권 거래 서비스를 실질적으로 구현하였다. 본 논문에서 구현한 증권 정보 서비스는 각종 지수관련 정보(종합 주가 지수, 코스닥 지수, 나스닥 지수, 업종별 지

수)를 제공하고 있으며, 종합거래소에 상장된 850여개의 종목에 대한 현재가 정보를 제공하고 있다. 또한 회원의 경우, 주식을 매도 또는 매수할 수 있는 서비스가 구현되어져 있으며, 매도/매수에 따른 실시간 계좌 정보 변경 기능을 구현하였다.

서비스 시나리오 변경 시 실시간 적용 및 운용자 편의를 위하여 운용자에게 웹을 경유한 시나리오 수정 및 갱신 기능을 제공하였다. 기존방식의 유사 서비스를 제공하는 시스템의 경우, 서비스 흐름 및 제공 콘텐츠가 서버에 하드코딩 되어져 있으므로, 이를 변경하기 위해서는 추가적인 프로그램의 코딩 및 컴파일, 서버로의 전송등을 필요로 한다. (그림 6)은 본 논문에서 구현한 음성 인식 증권 거래 시스템의 구성도다.

(그림 6) 구현한 음성 인식 증권 거래 시스템 구성도

3.1 구성환경

본 논문에서 구현한 음성 인식 증권 거래 서비스 시스템은 Dell Precision Workstation(CPU : Xeon 2.4GHz, Memory : 1GB)과 4채널을 가지는 Dialogic 41JCT/LS로 구성 되어 있다.

운영체제로써 Windows2000 Professional을 사용하였으며, 기타 소프트웨어로는 웹 서버 구동을 위한 IIS(Internet Information Services)와 데이터베이스 처리를 위한 Microsoft Access를 사용하였다. 또한, 게이트웨이는 KT의 HUVOICE 1.0을 사용하였으며, 시나리오 작성 시 VoiceXML과 ASP 언어를 사용하였다.

3.2 서비스 흐름도

전체적인 서비스 흐름도는 (그림 7) 및 (그림 8)과 같다. 본 논문에서 구현한 시스템의 경우 초기 메뉴로 여러 종류의 서비스를 제공하고 있으며, 이중 증권정보 서비스를 이용할 경우 사용자가 첫 안내 멘트 중간에 “증권정보”라 발성하면 해당 서비스로 이동하게 된다.

이외에도, 본 논문에서 구현한 시스템의 서비스로는 대학생과 대학원생들을 위한 강의안내 서비스, 공지사항 서비스, 전화번호 안내를 위한 연구실 및 연구원 연락처 안내

서비스 등을 제공하고 있다. 또한, 전화망을 경유한 음성신호의 특징 분석을 위하여 데이터베이스로 구축할 수 있는 음성 수집 서비스가 구현 되어져 있으며 현재 말로하는 음성게임 서비스가 구현 중에 있다. 이런 여러 서비스가 제공될 수 있는 이유는 시나리오 내용을 시스템과 독립적으로 구현함으로써, 웹서버에 시나리오 문서를 분리하여 저장할 수 있고 웹을 경유한 Link(URL)로 타 서비스와 연결이 가능하도록 구현하였기 때문이다.

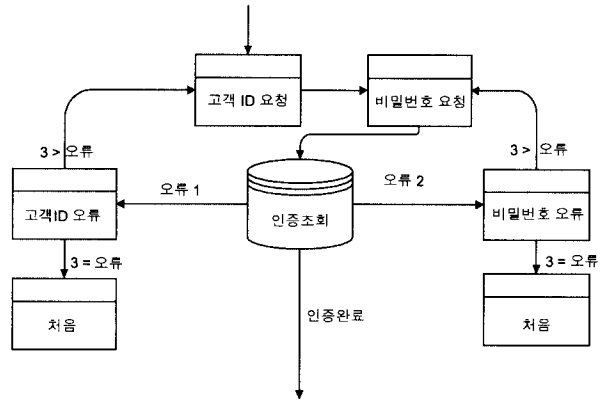
(그림 7) 현 시스템에서 제공되는 서비스 시나리오 블록도

현재 음성 인식 증권 정보 서비스에서 비회원을 위한 서비스로는 시세안내인 각종 지수정보(종합 주가 지수, 코스닥 지수, 나스닥 지수, 업종별 지수)를 제공하고 있으며 회원은 로그인 후 시세 안내 및 주식 매매, 그리고 매매에 따른 계좌정보의 조회가 가능하다. 주식 거래 시 주문 내역과 체결 내역을 조회할 수 있는 체결 안내 서비스도 제공하고 있다. 계좌정보에는 사용자의 예수금과 잔고, 그리고 보유한 주식의 수량 및 그 주식에 대한 수익률 정보를 제공하고 있다.

(그림 8) 증권 거래 시나리오 블록도

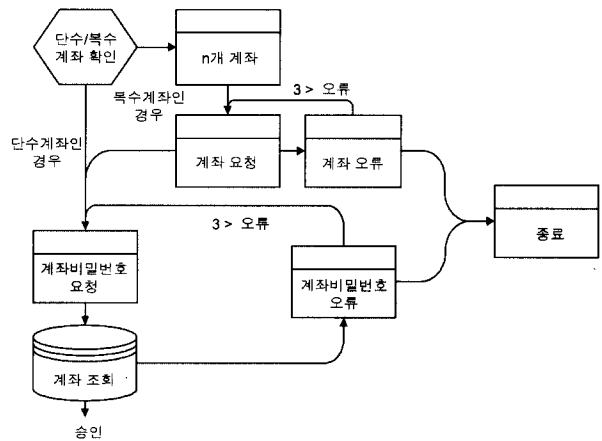
(그림 8) 및 (그림 9)에서 보여지는바와 같이 회원일 경우 로그인 시 7자리 회원번호와 비밀번호를 입력하면 회원 데이터베이스에서 인증을 조회한다. 또한 금융거래의 안전을 위하여, 3번 이상 연속하여 회원번호와 비밀번호의 입력에 오류가 있게 되면 (그림 9)에서 보여 지는 바와 같이 초기 메뉴로의 이동 또는, 서비스를 일정시간 동안 중지하는 기

능도 제공하고 있다(사용자별 개인설정에 따른 서비스 제공). 개략적인 로그인 흐름도는 (그림 9)와 같다.



(그림 9) 로그인 흐름도

서비스 로그인 후 회원이 계좌에 대한 정보를 확인할 경우 전체적인 흐름도는 (그림 10)과 같다. 사용자의 보유 계좌 수에 따른 선택적인 서비스 제공이 가능하며, 각 계좌별 비밀번호의 설정이 가능하다. 사용자가 계좌 정보를 요청할 경우 단수 계좌를 가지고 있는지 복수 계좌를 가지고 있는지 확인 후 사용자가 선택한 계좌의 비밀번호가 3번 이상 오류가 발생하면 자연스럽게 서비스는 종료되어진다.



(그림 10) 계좌 정보 흐름도

3.3 실험 및 결과

현재 본 논문에서 구현한 시스템은 850여개의 상장 종목에 대하여 종목명, 종목당 전일가격 및 현재가격을 데이터베이스로 구축하였으며 회원 관련 정보도 데이터베이스로 구축하였다. 구축되어진 데이터베이스와 VoiceXML 문서의 연동을 위하여 Script 언어 중 하나인 ASP를 사용하였다. 또한, VoiceXML 파일과 ASP 파일들을 서비스 항목별로 세부적으로 나누어 구현했기 때문에 서비스 제작자 이외의 VoiceXML 문법을 아는 일반 사용자도 쉽게 문서를 수정 및 갱신하기 쉽도록 구성하였다.

(그림 11)은 850여개의 상품 종목을 발성하여 인식한 결과를 나타낸다. 인식 실험은 시스템이 구축 되어진 학교 내부의 내선과 외선 등의 유선망과 이동전화를 이용한 무선망을 이용하였으며, 발성 화자는 남성 6명, 여성 4명으로 구성하였다. 실험 환경은 사무실 환경이며 인식기는 VoiceXML 게이트웨이에서 제공되는 인식기를 사용하였다.

발성화자들은 미리 교부되어진 A4용지 9매 분량의 상품 목록을 보고 발성하였으며, 1Page부터 순차적으로 발성토록 하였다. 남성의 경우 1차 발성해서 얻은 인식률은 95.65%였으며, 1차 발성시 오인식되었을 경우 3번까지 연속 발성토록 하였다. 남성의 경우 3회 발성에 따른 전체 인식률은 98.72%의 인식률을 보였다. 여성은 1차 발성시 92.94%의 인식률을 보였으며, 남성과 동일한 방법으로 3차 발성 실험을 하였으며, 이에 따른 전체 결과는 98.84%의 인식률을 보였다.

(그림 11)에서 보여지는 바와 같이 첫 페이지의 인식률은 다른 페이지의 인식률보다 다소 낮은 경향을 나타내었으며 이는 발성 화자가 인식기에 적응해 나가는 과정을 나타낸 것으로 분석되어진다.

(그림 11) 855종목에 대한 Page별 남성과 여성의 인식률

(그림 12) 실시간 정보 입력창

(그림 13) 안내 정보 입력창

(그림 14) 시황 속보 입력창

(그림 12), (그림 13) 및 (그림 14)는 개발자가 아닌 운영자가 실시간으로 필요한 정보를 웹 페이지에 입력하여 서비스 시나리오를 수정 및 갱신할 수 있도록 만든 웹 페이지이다. 운영자는 웹 페이지를 통해 시간과 장소에 구애받지 않으며 자유롭게 시나리오를 변경할 수 있다. 여기에 등록된 정보는 데이터베이스에 저장되고 해당 VoiceXML 파일을 불러올 때 데이터베이스에 저장된 정보를 사용자에게 들려주게 된다. 운영자가 아닌 일반인들의 정보 수정 및 갱신을 막기위해 관리자 비밀번호를 설정해 놓았다.

3.4 문제점 및 해결방안

ECMAScript로 다양한 표현식과 연산자, 내장함수들을 이용할 수 있는데 현재 제공되는 VoiceXML 게이트웨이로는 한계가 있었다. 따라서 현재 구현된 서비스는 서버측 스크립트 언어인 ASP를 사용하였다. ECMAScript 언어대신 서버측 스크립트 언어를 사용하게 되면 단점이 있다. 클라이언트인 VoiceXML 게이트웨이가 VoiceXML 문서를 요청하는 경우 앞서 만들어진 VoiceXML 문서를 캐시에 저장하고 있다가 필요할 때마다 사용할 수 있다. 그러나, 서버측 스

크립트 언어를 사용하여 VoiceXML 문서를 동적으로 생성시키면 이 문서는 클라이언트인 VoiceXML 게이트웨이에서 캐싱할 수 없게 된다. 서버측 스크립트 언어의 해석은 웹 서버에서 이루어지기 때문이다. 즉, VoiceXML과 ECMAScript만으로 처리할 수 있는 일은 VoiceXML과 ECMAScript를 사용하여 구현하는 것이 응답속도의 측면에서 더 유리하다.

주식을 매도 또는 매수시 음성으로 거래할 경우 후보수가 9,999,999개나 선정을 해야 한다. 이렇게 선정할 경우 후보수 증가에 따라 인식 시간이 많이 걸리며 인식률도 떨어지게 된다. 이런 문제점은 음성 문법에서 제공하는 핵심어 검출 방법을 응용하여 음성 문법을 구현하였다. (그림 15)는 주식 금액을 음성으로 발성할 경우 인식할 후보에 대해 구현한 음성 문법이다.

```
#ABNF 1.0 ISO-8859-1 ;

root $data_class ;

$Num1 = 이(2 ; ) | 삼(3 ; ) | 사(4 ; ) | 오(5 ; ) | 육(6 ; ) | 칠(7 ; ) |
팔(8 ; ) | 구(9 ; ) ;
$Num10 = 십(10 ; ) | 이십(20 ; ) | 삼십(30 ; ) | 사십(40 ; ) |
오십(50 ; ) | 육십(60 ; ) | 칠십(70 ; ) | 팔십(80 ; ) |
구십(90 ; ) ;
$Num100 = 백(100 ; ) | 이백(200 ; ) | 삼백(300 ; ) | 사백(400 ; ) |
오백(500 ; ) | 육백(600 ; ) | 칠백(700 ; ) | 팔백(800 ; ) |
구백(900 ; ) ;
$Num1000 = 천(1000 ; ) | 이천(2000 ; ) | 삼천(3000 ; ) | 사천(4000 ; ) |
오천(5000 ; ) | 육천(6000 ; ) | 칠천(7000 ; ) |
팔천(8000 ; ) | 구천(9000 ; ) ;
$Num10000 = 만(10000 ; ) | 이만(20000 ; ) | 삼만(30000 ; ) |
사만(40000 ; ) | 오만(50000 ; ) | 육만(60000 ; ) |
칠만(70000 ; ) | 팔만(80000 ; ) | 구만(90000 ; ) ;

$Num = [$Num1000]([ $Num100]([ $Num10]([ $Num10000 ] $Q1))) ;
$Num2 = [$Num1000]([ $Num100]([ $Num10] $Q1)) ;
$Jua = 원( ; ) ;

public $data_class = [$Num]([ $Num2]$Jua | $Q1) ;
```

(그림 15) 주식 금액 관련 음성 문법

이렇게 하여 음성 인식된 결과값은 문자로만 리턴되었다. 문자로 리턴되는 경우 계좌 정보에 들어가는 예치금 및 수익률을 계산하기가 어렵다. 이러한 문제점은 ASP 언어를 이용하여 결과값인 문자를 숫자로 변경하여 계좌에 관련된 사항에 대해 연산할 수 있도록 구현하였다. (그림 16)은 문자를 숫자로 변경하는 프로그램의 일부분이다.

```
<? xml version = "1.0" encoding = "EUC-KR" ?>
< vxml version = "1.0" >

< %
.
.
.
```

```
arrTemp = split(jusu, " ")
leng = ubound(arrTemp)
data = Cint(arrTemp(0))
for i = 1 to leng
  if arrTemp(i) <> "주" then
    if dd > Cint(arrTemp(i)) then
      data = data + Cint(arrT(i))
    else
      data = data × 10000
      data = data + Cint(arrT(i))
    end if
  end if
next

DBpath = server.mappath("siseDB.mdb")
.
.
.
< form id = "true" >
  < field name = "ans" type = "boolean" >
  < prompt bargein = "true" >
    < % = stock %> 주식은 현재 < % = current %> 원으로 전일대비
    < % = current - before %> 원 < % = str %>, < % = data %>
    주 주문하셨습니까?
  </ prompt >
  < grammar > 예 | 아니오 </ grammar >
  < filled >
  < if cond = "ans == '예'" >
  < goto next = "../process.vxml" />
  < elseif cond = "ans == '아니오'" />
  < goto next = "../mesu.vxml" />
  .
  .
  .
```

(그림 16) 리턴된 문자를 숫자로 변경하는 프로그램 일부분

현재 구현된 증권 거래 서비스는 동시 4채널을 사용할 수 있으며, 채널 수는 접속자 수에 따른 선택적 확장이 가능하며, 확장 시 소프트웨어의 변경 없이 동작 가능하도록 구현하였다. 초기 운영시 도출된 문제점으로는 접속자 증가에 대한 부하를 들 수 있으며, 본 논문에서는 초기 시스템의 보완을 위해 부하의 분산을 유도하였으며, 해결 방안으로 해석기와 인식기, 합성기를 각각 다른 서버로 구현, 운용하였다.

사용자 입력으로부터 응답에 대한 출력까지 단계를 C나 C++등의 언어로 하드 코딩된 기존의 음성 서비스 방식과 비교하면 기존 방식보다 VoiceXML을 이용한 음성 서비스 방식의 처리단계가 더 많다. 또한, 문서 서버인 웹 서버를 이용하기 위해 인터넷 망을 경유할 경우 인터넷 망의 트래픽 상황에 따라 서비스가 제공되어지는 속도가 느려지는 경우도 발생하였다. 이러한 문제점의 효율적인 대처를 위하여, 본 논문에서는 시간 지연이 길어질 것으로 예상되는 부분에는 배경음악을 문서에 삽입, 활용하여 사용자들이 느끼는 지루함 및 서비스 불만사항을 경감하여주도록 노력하였다. 이 문제에 대한 또 다른 해결 방법으로써 웹 서버쪽 콘텐츠 만료(content expiration) 옵션도 사용하였다. 콘텐츠

만료 옵션을 사용할 경우, 사용자에게 캐쉬된 페이지의 정보를 가져다줄지 서버에 업데이트된 페이지의 정보를 요청할지 결정할 수 있다.

원할한 서비스 제공을 위하여 사용자에게 오디오 파일을 재생하여 출력하는 경우, 오디오 파일의 형식 또한 서비스 품질에 영향을 미치게 된다. 현재 사용되어지고 있는 디지털 전화 교환기는 모두 8KHz로 샘플링된 8bit μ -Low PCM (Pulse Coded Modulation) 또는 A-Low PCM 데이터를 주고받는다. 따라서 16bit 16KHz의 높은 양질의 오디오 파일을 VoiceXML 문서에 지정한다고 하더라도 전화망을 거쳐서 사용자에게 전달되는 오디오 음악의 품질은 8KHz 8bit PCM 데이터이기 때문에 고급 오디오 파일을 사용하는 것은 파일 크기의 증가 및 응답시간의 지연만이 발생하며, 실질적으로 사용자가 느끼는 음질의 개선에는 도움이 되지 않는다.

4. 결 론

인터넷의 활성화로 매우 다양하고 많은 정보가 데이터베이스화 되어지고 있으며, 데이터 베이스화 되어진 정보는 사용자의 요구에 따라 24시간 교류되고 있다. 하지만, 사용자가 그 정보를 이용하려면 컴퓨터, 전용선과 같은 하드웨어 장치와 모니터등의 출력장치 제어를 위한 추가적인 소프트웨어를 필요로 한다.

현재 서비스 업체에서 제공하고 있는 전화 음성 서비스는 기존의 구축되어진 인터넷상의 정보를 활용하기가 쉽지 않고, 이러한 정보들을 전화 음성서비스로 전환하기에 시스템 구조상 쉽지 않았다. 또한 시스템과 관련된 전문적 지식이 없을 시에는 서비스 내용을 수정 또는 갱신하기가 어려웠다.

본 논문에서는 VoiceXML을 이용한 응용 서비스 중 음성 인식 증권 거래 서비스를 구현하였다. 구현하면서 발생한 문제점으로는 첫 번째, 음성을 이용하여 주식의 매매가 이루어질 경우 후보수 증가로 인해 인식을 저하와 인식시간 지연이 문제되었다. 이를 개선하기 위해 핵심어 검출방법을 응용하여 후보수를 45개로 줄이고 인식률과 인식속도를 개선하였다. 두 번째, 현재 보유한 VoiceXML 게이트웨이는 음성을 인식하여 결과 값을 문자로만 리턴하는데 이는 수익률 및 예치금을 계산할 때 문제점이 되었다. 이러한 문제점은 ASP 언어를 이용하여 리턴되는 문자를 숫자로 변경할 수 있도록 프로그램하였다. 또한 개발자가 아닌 서비스 제공업자에게 서비스 시나리오를 수정 및 갱신이 용이하도록 웹 페이지를 통해 수정 및 갱신 기능을 제공하였다. 인터넷망이 연결되는 곳이라면 장소와 시간에 구애받지 않고 서비스 제공업자가 직접 시나리오를 수정 및 갱신할 수 있다.

VoiceXML을 이용하여 응용 서비스를 구현하게 되면 웹의 하이퍼링크와 같은 기능으로 원하는 타 서비스업체와 연결할 수 있다. 사용자들은 전화망 포털 서비스의 전화번호만 알면 컴퓨터 없이 전화기로 손쉽게 원하는 정보를 획득할 수 있게 된다.

향후 연구방향으로는 미리 정의된 순서에 따라 사용자와 컴퓨터가 대화하는 수동적인 다이얼로그 모델인 기계 주도형 단방향 구조(Machine Directed Form)에서 사용자와 컴퓨터 모두 대화의 진행을 능동적으로 변경할 수 있는 다이얼로그 모델인 상호 주도 구조(Mixed Initiative Form)로 서비스 시나리오를 개선해야 할 것이다. 그리고, 핵심어 검출법을 이용하여 고립 단어 인식에서 대화체 음성 인식이 이루어 지도록 개선되어야 할 것이다. 또한, VoiceXML을 이용한 서비스 구현 시 발생할 수 있는 문제점과 효율적인 대처방안의 교류가 보다 확대되어져야 할 것이다.

참 고 문 헌

- [1] VoiceXML Forum, Voice eXtensible Markup Language, <http://www.voicexml.org/specs/VoiceXML-100.pdf>.
- [2] Goose, S., M. Newman, C. Schmidt and L. Hue, "Enhancing Web accessibility via the Vox portal and a Web-hosted dynamic HTML voxML converter," Computer Networks Vol.33, pp.583-592, 2000.
- [3] 김학균 외, "VoiceXML 기반 음성인식시스템을 이용한 서비스 개발", 말소리, 제43호, pp.113-124, 2002.
- [4] Bruce Lucas, "VoiceXML for Web-based Distributed Conversational Applications," Communications of the ACM, 2000.
- [5] Quiane Ruiz, J. A. Manjarrez Sanchez, J. R., "Design of a voice XML gateway," ENC '03, pp.49-53, Sept., 2003.
- [6] Quiane Ruiz jorge A., "Design and Implementation of Voice XML Gateway," Center for Computing Research-IPN, Mexico, 2003.
- [7] W3C DOM WG, "Document Object Model(DOM)," <http://www.w3.org/DOM/>, W3C, 2002.
- [8] <http://www.w3.org/TR/2004/PR-DOM-Level-3-Core-20040205/>, W3C, 2003.

조 창 수

e-mail : wildzzang@empal.com

2002년 호원대학교 컴퓨터공학과(학사)

2002년~현재 성균관대학교 정보통신공학부 석사과정

관심분야 : 음성인식 및 합성, VXML

신 정 훈

e-mail : only4you@chol.com

1992년 성균관대학교 전자공학과(학사)

1994년 성균관대학교 대학원 전자공학과
(공학석사)

2002년 성균관대학교 대학원 전기 전자
및 컴퓨터공학과(공학박사수료)

1994년 (주)SKC 중앙연구소 연구원

1995년~2002년 (주)DACOM 종합연구소 주임연구원

2003년~현재 인덕대학 정보통신전공 겸임 교수

관심분야 : 유무선 통신, 음성 인식 및 합성, 오감융합, CTI

홍 광 석

e-mail : kshong@skku.ac.kr

1985년 성균관대학교 전자공학과(학사)

1988년 성균관대학교 전자공학과
(공학석사)

1992년 성균관대학교 전자공학과
(공학박사)

1990년~1993년 서울보건전문대학 전산정보처리과 전임강사

1993년~1995년 제주대학교 정보공학과 전임강사

1995년~현재 성균관대학교 정보통신공학부 교수

관심분야 : 오감인식, 융합 및 재현, HCI