

VoiceXML을 이용한 IVR 서버 설계 및 구현*

Design and Implementation of IVR Server Using VoiceXML

이 창 호** · 장 원 조*** · 강 선 미***
Changho Lee · Wonjo Jang · SunMee Kang

ABSTRACT

A new brilliant service using human-voice and DTMF (Dual Tone Multi Frequency) technique is expected nowadays in order to obtain valuable information on the internet more easily. VoiceXML (Voice eXtensible Markup Language) is the right choice that makes the new service possible. In this paper, the design and implementation of IVR (Interactive Voice Response) server using VoiceXML is described, where it connects with internet and IVR server efficiently. IVR server using VoiceXML is composed of two groups: VoiceXML document handling and VoiceXML execution. Scenario part of IVR server corresponds to VoiceXML document, the execution is performed by VoiceXML execution.

Keywords: VoiceXML, IVR, Speech Interface, DTMF, XML

1. 서 론

VoiceXML (Voice eXtensible Markup Language)은 1999년 미국 AT&T, IBM, 루슨트 테크놀러지 및 모토로라사에 의해 공동으로 제안된 대화형 마크업 언어로써 음성 어플리케이션 개발을 위해 고안된 XML 문서 형식의 일종이다. VoiceXML forum은 1999년 8월 VoiceXML version 0.9를 발표한 후 2000년 3월 이를 크게 보완하여 version 1.0을 정식 제안했다. VoiceXML 버전 2.0 working draft가 2001년 10월에 발표되었고, 또한 VoiceXML 버전 3.0이 계획되고 있다.[1] 현재 전 세계 382 개 이상의 회사가 VoiceXML 포럼에 참여하고 있다.

VoiceXML 서비스는 사용자의 음성과 전화의 DTMF 신호의 입력을 통해서 사용자가 원하는 정보를 검색 및 조회할 수 있도록 구성되었다. 이 서비스의 구현을 원활하게 하기 위해서는 다음과 같은 기능이 기본적으로 필요하다.[1][2]

* 본 논문은 부분적으로 한국과학재단 목적기초연구(R01-1999-00229) 지원으로 수행되었습니다.

** 보이스미디어텍(주)

*** 서경대학교 컴퓨터과학과

1. 음성 합성의 출력기능
2. 오디오 파일의 출력기능
3. 입력된 음성의 인식기능
4. 입력된 DTMF의 인식기능
5. 입력된 음성의 저장기능
6. 호 전환/해제와 같은 전화관련처리기능
7. 대화식 흐름에 의한 제어기능

현재는 위와 같은 기능을 부분적으로 구현한 VoiceXML 서비스가 이루어지고 있다. 그리고 VoiceXML 서비스를 시도하려는 업체들의 대부분은 IVR 서버를 구축했던 회사이다. 이는 VoiceXML 서비스는 IVR 서버의 구현에 필요로 하는 요소기술을 포함하고 있기 때문이다.

VoiceXML 표준이 소개되기 전에는 대부분의 IVR 서버는 특정한 서비스를 위한 극히 제한된 용도로 사용되어 왔다. 그리고 다른 서비스를 위해서는 시스템 개발자가 직접 시스템의 대부분을 수정해야만 했다. 그러나 VoiceXML 서비스는 VoiceXML 문서의 내용에 따라서 동적으로 서비스되도록 IVR 서버가 구성되어 있기 때문에 새로운 서비스를 위한 시스템 개발자의 수고를 필요로 하지 않는 장점을 가진다. 즉, VoiceXML 문서의 수정만으로도 서비스가 가능하게 된다. 그 이유는 서비스 시나리오와 사용자 인터페이스가 분리되어 있기 때문이다.

본 논문에서는 VoiceXML 문서에 의해서 작동되는 IVR 서버를 어떻게 효율적으로 구현할 것인가에 대한 구조 설계 및 구현에 대하여 기술하였다. 주요 내용은 다음과 같다.

첫째 VoiceXML 버전 1.0에서 제안된 VoiceXML에 대한 주요 내용에 대해서 기술하였다. 둘째 VoiceXML Interpreter에서 분석된 정보를 바탕으로 서비스가 가능하도록 VoiceXML 실행부에 대한 개발과 VoiceXML Interpreter 간의 정보전달을 위해 본 논문에서 제안한 방법 및 향후 어떠한 방식으로 효율적인 시스템을 구현해야 할 것인가에 대해서 기술하였다. 셋째 마지막으로 결론에 대해서 기술하였다.

2. IVR 시스템

IVR이란 Interactive Voice Response의 약자로, IVR을 직역하면 대화식 음성 응답 시스템이다. 서비스 이용 측면에서 보면 ARS와 유사하지만 IVR은 사용자가 요구하는 사항을 DB에서 검색한 후 음성합성 또는 음성조합의 방법을 이용하여 대화 형태로 이용자에게 제공되는 서비스이다.

IVR 시스템과 VoiceXML 시스템은 서비스의 내용상으로는 동일하다고 할 수 있다. 그러나 기존의 IVR 시스템은 표준화되지 않은 서비스인데 비해서 VoiceXML 시스템은 표준화된 서비스라고 할 수 있다. 이 의미는 VoiceXML 시스템은 어떠한 경우에도 VoiceXML 문서에 따라 서비스하는 것이 가능하다는 것이다. 이에 반해서 기존의 IVR 시스템은 VoiceXML 표준을 따르지 않아도 IVR 시스템의 구조에 따라 서비스가 가능하기 때문에 이종의 IVR 시스템과의 호환성의 문제가 발생할 수 있다.

VoiceXML 시스템은 IVR 시스템을 표준화하고 시스템 운영자가 서비스를 보다 편리하게 변경하는 것이 가능하도록 하였으며, 대표적인 서비스가 IVR 서비스라고 할 수 있다.

3. VoiceXML 구조[5]

VoiceXML은 오디오, 음성인식, DTMF, 음성입력, 전화 등 음성 기반의 사용자 인터페이스를 통해 웹 정보를 검색하는 어플리케이션 개발을 위해 고안된 XML 문서 형식의 일종이다. VoiceXML을 이용함으로써 기존의 음성서비스 제공업자들은 다양하고 쉽게 새로운 서비스를 제공할 수 있게 되며 음성 서비스 저작에 대한 업무를 분할하게 된다. 즉 음성 서비스 저작자는 음성 입출력에 관한 기술적인 문제에 대해서는 알 필요가 없고 VoiceXML 문서를 통해 서비스 자체에만 집중할 수 있게 한다. VoiceXML의 전체 구조는 그림 1과 같다.

3.1 VoiceXML Parser

VoiceXML 파서는 내부적으로 XML 파서를 내장해 VoiceXML 태그 내포 구조를 분석하고 이 구조가 VoiceXML DTD(Document Type Definition)에 알맞은 지의 유효성 검사를 한다. XML 파서의 경우 많은 공개 버전이 이용되고 있으며, 한글 처리와 웹 접근에 문제가 없도록 유의해야 한다. 에러에 대한 핸들링은 DOM(Document Object Model) 과 SAX(Simple API for XML) 모두 지원한다.

본 논문에서 MSXML DOM 파서를 사용하여 VoiceXML Interpreter를 구성하였다.

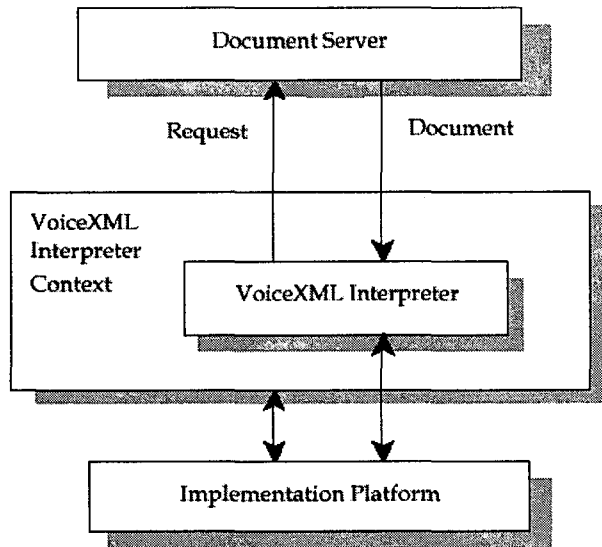


그림 1. VoiceXML Architectural Model

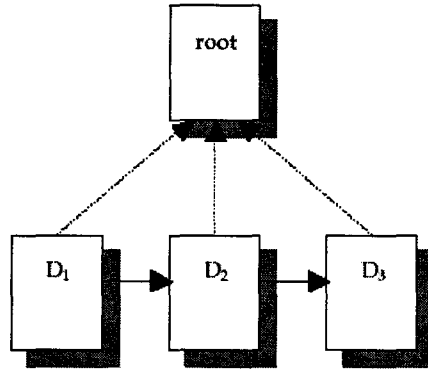


그림 2. 문서 전이 관계(D: Document)

3.2 VoiceXML Interpreter

VoiceXML 실행 환경의 가장 핵심적인 요소인 VoiceXML Interpreter는 문서를 적재하고 그 내용을 해석하고 실행하는 역할을 담당한다. VoiceXML에서 정의된 47 종의 element에 설정된 기능에 따라서 문서 실행의 순차적 흐름을 제어하고, 음성 입력력 내용을 결정해 음성 플랫폼에 필요한 명령을 내린다. 또한 문서 서버를 통해 필요한 자원을 다운 로드하거나 다른 문서로 전이하는 등 VoiceXML 문서 실행을 총괄적으로 제어하는 엔진 역할을 수행한다. 이러한 기능들은 10 개의 범주로 분류된다.[1] 내부적으로 XML 파서를 내장해 VoiceXML 태그 내포 구조를 분석하고, 제안된 구조가 VoiceXML DTD에 적절한 지 검사한다.

Document Server는 URI(Uniform Resource Identifier) 형태로 전송되는 VoiceXML Interpreter가 요청한 문서나 자원을 Interpreter에게 전송하는 역할을 담당한다. <submit> 구조에 의한 문서 전이 요청의 경우 새로운 VoiceXML 문서가 오디오나 grammar 구조의 경우 음성 파일이나 문법 파일이 전송된다. 문서 서버는 독자적인 서버를 구축하기보다는 기존의 웹 서버를 사용하는 것이 가장 간단하면서도 효율적이다. 웹 서버를 문서 서버로 이용할 경우 안정적인 프로토콜에서 전송이 가능해 개발 부담을 줄일 수 있다. 또한 CGI, ASP, PHP 등을 이용할 수 있어 사용자 입력에 따른 동적 문서 생성 및 웹 DB와의 연동으로 실시간 정보 반영이 가능해진다는 장점이 있다.

VoiceXML 어플리케이션은 그림 2와 같이 하나의 루트(root) 문서와 이 루트 문서를 참조하는 다수의 비 루트 문서로 구성된다. 루트 문서는 꼭 있어야만 하는 것은 아니고 어플리케이션에서 글로벌로 사용하고자 할 때 사용한다.

VoiceXML 최상위 구조인 <vxml> 구조는 하나 이상의 대화 구조를 갖는데 다이얼로그는 주로 form과 menu로 구성되며, 사용자와의 대화에서 사용되며 입력에 대한 결과를 출력한다. 그림 3과 같이 다이얼로그는 시나리오에 따라 한 문서에서 순차적으로 실행될 수도 있고 다른 문서의 다이얼로그로 전이했다가 다시 되돌아올 수도 있다. 그림에서 빗금 친 부분은 Document를 나타낸다.

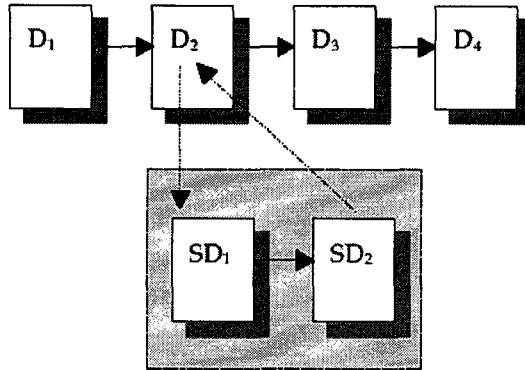


그림 3. 다이얼로그의 전이 (D: Document, SD: Sub Dialog)

VXML 변수와 grammar들은 다섯 가지의 scope를 가지고 있다. 처음 환경변수 등 사용자가 접속되어 있는 동안 모든 부분에서 이용되는 session scope(수화기를 들고 서비스를 받고 다시 수화기를 내려놓는 시점까지를 말하며, 이것은 전체 서비스를 말하는 것은 아니다)와 여러 문서에서 함께 사용될 수 있는 application scope, 그리고 하나의 문서에서 이용될 수 있는 document scope, 하나의 form이나 menu에서 사용되는 dialog scope, 하나의 element에서 사용되는 anonymous scope가 있다.

3.3 VoiceXML Interpreter Context

VoiceXML Interpreter Context는 시스템 제반 환경에 따라 VoiceXML Interpreter를 통제, 관리하는 역할을 담당한다. 특히 ARS 서비스나 음성 포털 지원을 위한 채널 인터프리터의 경우 사용자 접속 및 해제를 감지해 인터프리터 쓰레드의 할당, 가동, 중지 등을 제어하며 시작 문서의 지정, 환경 변수와 시스템 옵션의 설정 등을 관리한다. 다양한 환경에서 인터프리터의 환경 독립성을 극대화하기 위해 설정된 외곽 요소가 VoiceXML Interpreter Context이다.

VoiceXML 인터프리터를 구현하기 위해서는 Implementation Platform, VoiceXML Interpreter Context, VoiceXML Interpreter, Document Server가 필요하다. Implementation Platform은 VoIP 또는 기존 전화의 Call 신호처리, Event를 Interpreter나 VoiceXML Interpreter Context에 전달 입력 및 출력 처리한다. Interpreter Context은 User Input Monitoring, Session의 설정, 유지, 제어, Call 검출, 초기문서 수행, 환경 설정, Browsing 명령을 처리한다. Interpreter는 입력된 User Input으로 Request 생성과 Document Server에 전달 VoiceXML Parser를 이용하여 VoiceXML Tag 처리, VoiceXML Document 처리 및 전체 Dialog를 제어하고 Implementation Platform 제어한다. Document Server는 Request 처리하고 VoiceXML Document 생성 CGI에 CGI 및 웹 서버에 대응하는 기능을 수행한다.

3.4 Implementation Platform

본 논문에서는 VoiceXML 실행부에 해당하는 부분으로 음성인식기, 음성합성기, Telephony Board Control이 이에 해당한다.

VoiceXML 문서에 따른 동적 서비스를 위해서는 음성인식과 음성합성 등 음성서비스 지원

기능과 Telephony Board Control 부분은 전화망을 통한 서비스를 위해서 필수적으로 요구된다. 전화망을 통한 서비스가 아닌 경우에는 Telephony Board Control을 대신하는 다른 부분이 필요하다.

Implementation Platform은 음성서비스의 Platform에 따라서 Telephony Board Control 부분을 대체하는 새로운 부분이 필요하게 된다.

4. IVR 서버 설계 및 구현

4.1 VoiceXML을 이용한 IVR 서버 구조

본 논문에서는 그림 1에 나타낸 VoiceXML 스펙에 제안된 서비스 모델을 기본으로 시스템을 구현하였다. VoiceXML Interpreter의 음성 인터페이스는 음성 처리 엔진 기능을 포함한 전화 인터페이스 기능을 포함한다. 제안된 VoiceXML을 이용한 IVR 서버 구현시 시스템의 구조를 도식화하면 그림 4와 같다.

이 논문에 구현된 시스템에서는 VoiceXML Interpreter에서 해석된 정보를 INAP (Intelligent Network Application Protocol)[3]을 이용하여 VoiceXML 실행부와 통신함으로써 VoiceXML에서 나타난 문서의 내용에 따라서 서비스되도록 IVR 서버가 구현되었다.

시스템의 구성을 보면 크게 VoiceXML 문서 처리부, VoiceXML 실행부(음성인식기, TTS: Text To Speech, Speech Recorder, Telephony Board Control)로 나누어질 수 있다.

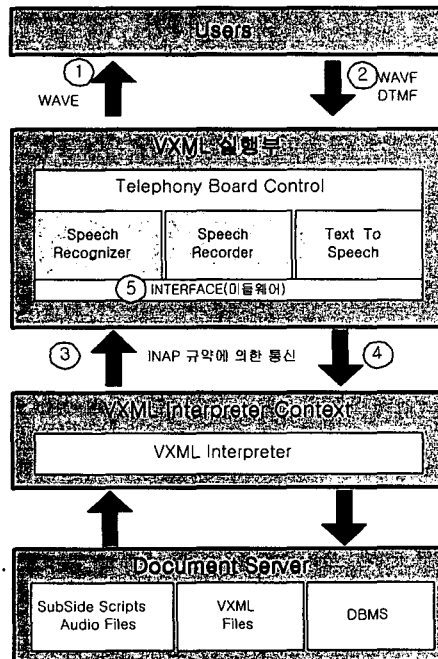


그림 4. VoiceXML을 이용한 IVR 시스템 구조

Telephony Board Control에서 주로 하는 일은 VoiceXML Interpreter에서 INAP 형식에 의해서 송신된 메시지를 수신하여 전화 인터페이스 보드가 해당하는 오퍼레이션에 따라서 작동하고, 처리 결과를 다시 VoiceXML Interpreter로 송신하는 것이다. 그리고 현재의 Telephony 보드의 상태에 따라서 VoiceXML Interpreter로 INAP 메시지를 송신하도록 구현되었다.

예를 들면, 어떤 음성을 출력하라는 오퍼레이션이 VoiceXML Interpreter에 의해서 해석될 경우에는 우선 출력할 문서에 대한 텍스트 정보와 음성을 출력하라는 오퍼레이션 정보를 동시에 Telephony Board Control로 보낸다. Telephony Board Control에서는 수신된 정보를 바탕으로 TTS에 의해서 텍스트 정보를 음성파일로 생성한 후에 해당 음성파일을 출력시켜 주면 원하는 서비스를 수행할 수 있다. 음성이 제대로 출력되었을 경우에는 출력이 제대로 실행되었다는 메시지를 VoiceXML Interpreter에 보내주어야 한다.

사용자의 입력을 원하는 서비스를 하는 경우에도 마찬가지로 사용자에게 안내 음성을 전달해주고 사용자가 DTMF 또는 음성을 입력했을 경우에 이 메시지의 결과 정보를 VoiceXML Interpreter에 전달해 주는 일을 한다.

Telephony Board Control에서는 VoiceXML Interpreter에서 송신하는 메시지의 내용에 따른 호 처리 및 안내 음성출력 및 사용자의 음성녹음 기능 그리고 DTMF 수집 등의 서비스가 VoiceXML 문서에 따라서 작동될 수 있도록 구현되었다.

이 논문에 구현된 시스템은 그림 4의 구조를 갖는다. (참고로 본 논문에서 구현한 시스템에서는 Document Server를 따로 분리하지 않았다.) 우선 VoiceXML Interpreter와 VoiceXML 실행 부와의 통신을 위해서 INAP 형식을 사용하였고, Telephony Board의 종류에 관계없이 보드를 제어할 수 있는 미들 웨어를 설계하였다. 그리고 VoiceXML 실행부에서 Telephony Board Control이 다른 VoiceXML execution 모듈을 제어할 수 있게 구성하였다. 이렇게 구성한 이유는 Telephony Board Control 부분이 사용자와 직접적으로 인터페이스하는 부분이기 때문에 시스템 구현상의 편리성 때문이다.

4.2 Telephony Board 독립적인 미들웨어 설계

1) 구현된 방법

구현된 IVR 서버는 다이얼로직 보드의 SDK를 사용하여 프로그래밍되어 있다. 만약 telephony board를 다른 회사의 보드로 교체하는 경우 해당 회사의 SDK를 사용하여 프로그래밍되어야 한다. 그러므로 보드를 직접 핸들링하는 부분을 각각의 기능별로 모듈화시켜서 보드의 교체에 따른 변경사항을 최대한 줄이도록 해야 한다. 현재 구현된 각 모듈은 표 1과 같으며, 각각의 채널별로 필요한 정보를 dxinfo 구조체로 선언하여 각 채널에 필요한 값을 저장할 수 있게 하였다.

표 1. 구현된 음성 모듈

함수명	기능	설 명
play()	음성출력모듈	해당 채널 번호와 파일 핸들 값을 입력. 함수 내에서는 다이얼로직에 관련된 구조체 정보를 설정하고 음성 출력과 관련된 함수 실행
play_digit()	음성출력 및 디지털입력	입력 가능한 디지털의 수는 dxinfo 구조체 내의 입력 디지털의 최대 개수에 해당하는 정보 입력 설정
record()	음성녹음모듈	해당 채널 번호와 파일 핸들 값 입력. 모듈 내에서는 다이얼로직에 관련된 구조체 정보를 설정하고 음성 녹음과 관련된 함수 실행
play_recogn()	음성출력 및 음성입력	음성을 출력하면서 실시간 끝점루틴 알고리즘을 이용하여 음성입력

궁극적으로는 CTI 표준인 S.100에 의해서 표준화된 API를 사용하여 IVR 서버를 구축한다면, 하드웨어 보드에 교체에 따른 여러 가지 문제점을 효과적으로 막을 수 있는 최선의 방법이라고 할 수 있다.

2) 향후 구현할 방법

하드웨어 Platform에 독립적으로 작동할 수 있는 IVR 서버를 구축하기 위한 최선의 방법은 ECTF (Enterprise Computer Telephony Forum)에 의해서 제안된 S.100, S.200에 따른 것이라 판단된다.[4] S.100 스펙에는 computer telephony application 개발을 위한 구조와 API가 기술되어 있다.

S.100과 S.200에 대한 내용은 다음과 같다.

- S.100은 computer telephony service를 액세스할 수 있는 어플리케이션을 제공하는 C로 작성된 프로그램 인터페이스이다.
- S.100 framework에는 호 처리 하드웨어, 스위치 구조, 그리고 이식성 있는 어플리케이션을 작성할 수 있는 토폴로지 환경설정 등에 대한 구현항목들이 기술되어 있다.
- S.200은 전송 계층과 독립적인 S.100을 보완하는 어플리케이션 프로토콜이다.
- S.200 스펙에는 S.100을 보완하는 클라이언트/서버 어플리케이션 계층이 정의되어 있다.
- S.100으로 구현된 Computer Telephony Service는 어디에서나 S.200 프로토콜을 사용하여 해당 자원을 사용할 수 있다.
- S.200은 S.100 클라이언트와 ECTF S.100 서버 사이의 상호작용을 가능하게 하므로 독립적으로 개발하여 시스템을 구성할 수 있다.
- S.100, S.200을 통해서 시스템 소유자가 얻을 수 있는 특징과 이점은 다음과 같다.
 - 특별한 하드웨어 제품에 종속적인 부분을 제거하기 때문에 하드웨어 독립적이다.
 - 어플리케이션이 모듈성을 가진다. 각각의 어플리케이션, 서버, 클라이언트별로 업그레이드할 수 있다.
 - 클라이언트/서버 구조를 갖는다.
 - Application layer protocol로 S.200을 사용한다.

현재의 시스템에서 Telephony 보드 독립적인 미들웨어가 S.100에 해당하고, VoiceXML Interpreter와 VoiceXML 실행부 간의 통신부분이 S.200에 해당한다.

4.3 Telephony Board Control의 세부 기능[5]

1) 호 처리(Call Process)

호의 connection과 disconnection에 대한 모든 기능과 호 전환을 위한 기능을 말한다. 참고로 VoiceXML 문장이 종료되기 전에 사용자가 전화를 끊을 경우에는 호를 disconnection 할 수 있는 메시지를 VoiceXML Interpreter에게 자동적으로 송신할 수 있도록 설계해야 된다. 그리고 현재의 전화의 상태 정보를 알아야 된다. 예를 들면 ON_HOOK인 경우는 현재 전화를 받을 수 있는 상태임을 나타내고, OFF_HOOK인 경우는 외부에 전화를 하기 위한 대기상태이거나 현재 사용자의 호가 연결된 상태임을 나타낸다.

2) 음성 출력(PA : Play Announcement)

사용자에게 음성을 출력하는 서비스를 위한 기능이다. VoiceXML Interpreter에서 송신하는 INAP 메시지의 정보에 따라서 text 정보를 받아서 TTS를 통해서 음성 파일을 만들고, 사용자에게 해당 음성을 출력하는 역할을 한다.

3) 사용자 음성 녹음(Speech Recorder)

전화를 통해서 말하고 있는 사용자의 음성을 녹음할 수 있는 기능을 말한다. 녹음된 음성이 그대로 필요할 수도 있고, 다음에 나오는 여러 가지의 서비스를 선택하기 위해서 음성인식을 위해 녹음하는 경우가 있다. 만약 음성인식이 필요한 경우라면 이 음성 메시지를 음성인식에 보내준 다음 음성 인식 결과를 받아서 해당되는 처리를 해야 된다.

4) DTMF (Dual Tone Multi Frequency) 수집

사용자가 서비스의 안내 음성에 따라서 전화기의 버튼을 선택할 경우 선택한 DTMF가 무엇인지를 알아내는 기능을 한다. 이 기능은 전화선 처리 보드의 API 함수를 통해 간단히 구현될 수 있다.

5) Barge-In 기능

안내 음성이 나오는 중이라도 언제든지 사용자의 음성 녹음이 가능하도록 하는 기능이다. 즉 매번 반복되는 안내 음성을 들을 필요가 없이 원하는 서비스를 하기 위한 기능이다. 참고로 Barge-In 기능과 상대적인 Prompt 기능이 있는데 이 기능은 사용자가 음성을 녹음하는 시점이 정해져 있어서 안내 음성이 나오는 중에 음성을 녹음할 수 없다.

6) 끝점 처리 기능(Endpoint Detect)

신뢰성 있는 IVR 서버를 구현하기 위해서 현재 Telephony Board에서 음성을 검출하기 위해서 지원되는 여러 기능에 추가하여 실시간 음성 검출을 위한 끝점처리 알고리즘을 사용하였다.

7) VoiceXML을 IVR 서버에 적용하기 위한 보완 기능

VoiceXML 문서를 통해 해석된 정보를 바탕으로 Telephony Board Control이 작동되는 것이 원칙이나 VoiceXML 인터프리터의 한계에 의해서 IVR 서버를 구축하는데 문제가 발생하는 경우가 있다. 즉, VoiceXML 스펙에 따라서 완벽하게 VoiceXML Interpreter를 구현하지 못한 경우에는 Telephony Board Control 부분이 이를 보완하기 위한 기능을 추가해야 한다.

예를 들어 VoiceXML Interpreter가 DTMF 입력에 관한 스펙을 모두 수용하지 못할 경우에, 사용자의 패스워드를 DTMF를 통해 입력받겠다고 할 경우 만약 패스워드가 4 자리에서 6 자리 사이로 정해져 있다면 DTMF를 사용자가 4자리를 입력할지 아니면 5 자리를 입력할지 6 자리를 입력할 지를 모르기 때문에 패스워드를 입력한 후 “#”를 입력하라는 메시지와 같은 보완 루틴을 추가해야 한다.

이와 같이 현재의 VoiceXML Interpreter의 해석 능력에 따라서 VoiceXML Interpreter가 처리해야 할 내용과 Telephony Board Control에서 처리할 부분을 명확히 결정해서 IVR 서버를 구축해야 한다.

물론 VoiceXML 버전 1.0을 완벽하게 만족하는 VoiceXML Interpreter인 경우에는 IVR 서비스를 위해서 보완하는 기능이 필요하지 않는다.

DTMF의 입력에 관한 보완 사항은 원칙적으로 VoiceXML 버전 1.0에서 APPENDIX F.에 수록되어 있는 방법에 따라 VoiceXML Interpreter와 Telephony Board Control을 구성하면 아무런 문제가 발생하지 않는다. APPENDIX F.에 수록되어 있는 방법은 TIMEOUT 방법을 사용하여 일정시간동안 입력이 없을 경우 그전까지 입력된 DTMF 버튼을 사용자의 입력으로 보는 것이다. 그 외의 여러 가지 경우가 나타나 있다. VoiceXML Interpreter에서 VoiceXML의 스펙에 있는 모든 내용을 처리하지 못하는 경우에는 우선 시스템 전체의 구성을 위해서 Telephony Board Control이 보완하여 처리하는 것이 효율적이다.

4.4 음성 인식기와 음성 합성기의 구성

음성 인식기와 음성 합성기는 SDK(Software Development Kit) 형태로 구성하여 손쉽게 해당하는 기능을 구현할 수 있도록 되어 있다.

음성 인식기는 VoiceXML Interpreter에 의해서 추출되어진 음성 인식 대상 단어에 대한 사전을 만든다. 그리고 나서 Telephony Board에 의하여 사용자의 음성을 입력받고, 해당 음성을 음성 인식기에 보내면 인식 결과가 해당 단어의 Text 값으로 출력되도록 구성되어 있다.

음성 인식기에서 사전을 만들어 해당하는 사전에 대하여 인식을 하는 것은 음성인식 오류와 인식 속도를 줄이기 위한 방법이다. 음성 합성기는 VoiceXML Interpreter에 의해서 추출되어진 Text 정보를 음성 합성기의 입력으로 넣어 주면, wave 파일이 출력된다. 출력된 wave 파일을 Telephony Board에 의해서 사용자에게 출력되도록 구성되었다.

음성 인식기 및 음성합성기에 사용된 API 함수를 표 2와 같다.

표 2. 구현된 음성 인식기 및 음성합성기 API들

함수명	기능 설명
VMTAsrInit ()	음성 인식기 초기화한다.
VMTAsrMakeDic ()	사전 파일을 이용하여 사전에 해당하는 단어의 발음 사전을 만들어 낸다.
VMTAsrSetSpeed ()	매개변수는 1부터 100까지의 값을 가질 수 있고, 인식 속도를 나타낸다.
VMTAsrStart ()	음성파일을 입력하여 단어 사전에서 해당되는 인식 결과를 Text 값으로 리턴해 준다.
VMTAsrClose()	음성인식기를 종료한다.
TtsInitialize()	음성 합성기를 초기화 한다.
TtsSelectEngine()	음성 합성기에 사용 할 엔진을 선택한다.
TtsGetVoiceInfo()	음성 합성기에서 특정 언어에 사용할 유용한 음성에 관한 정보를 검색한다.
TtsGetCtxModeInfoEx()	음성 합성기에서 전처리를 위해서 사용하는 TTSCXTXLIST 구조체의 상세한 정보를 검색하는 역할을 한다. 단, 이 함수가 호출되기 전에 음성합성엔진이 선택되어야 한다.
TtsSetSpeechMode()	음성합성기에 사용될 기본적인 매개변수(볼륨, 화자 타입, 피치 등)를 설정한다.
TtsEnableNotify()	메시지에 의해서 현재의 상태를 공지하는 역할을 한다.
TtsSetFreq()	음성 합성기에 의해서 생성될 음성의 샘플링 수를 결정한다.
TtsGenPcmFile()	Text 값을 특정 형식의 PCM 데이터로 생성해준다. 즉, 이 함수는 text를 음성파일로 출력해준다.

4.5 INAP 통신

구현된 시스템에서는 VoiceXML Interpreter와 VoiceXML 실행부 간의 통신을 위하여 INAP(Intelligent Network Application Protocol)을 사용하였다.

INAP이란 효율적인 서비스 제어를 목적으로 망의 기능이 여러 개의 시스템으로 분산되는 구조를 갖는 지능망에서는 시스템 간에 긴밀한 정보 교환 방안을 고려하여 지능망 요소들간의 상호작용을 위하여 ITU-T(International Telecommunication Union-Telecommunication Standardization Sector)에서 권고한 지능망 응용 프로토콜을 말한다. 그림 5은 VoiceXML 인터프리터와 VoiceXML 실행부 간의 통신 절차에 대하여 나타낸 것이다.

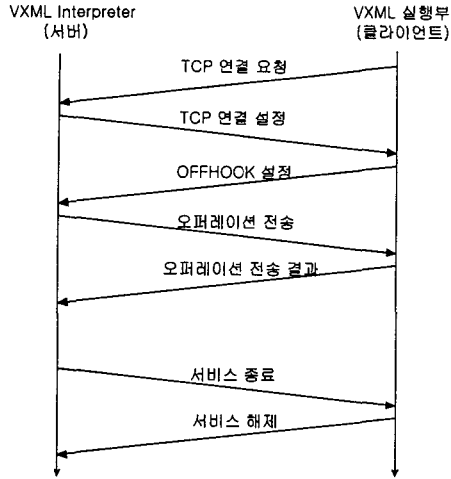


그림 5. 통신절차

5. 결 론

본 논문에 구현된 시스템은 VoiceXML 스펙 1.0을 기본으로 하였으며, 기존의 IVR 서비스에 적용하여 그 기능을 확인하였다. 구현된 시스템은 크게 VoiceXML 문서 처리부, VoiceXML 실행부로 나눌 수 있으며, VoiceXML 문서 처리부에서 해석된 정보를 VoiceXML 실행부의 Telephony Board Control이 담당하여 처리하도록 하였다. VoiceXML을 이용한 IVR 시스템에서는 실질적으로 VoiceXML 인터프리터에 해석된 정보를 바탕으로 사용자에게 직접적으로 서비스해 주는 부분을 메인으로 구성하고, 그 외의 음성 인식/합성 모듈 등은 별도의 프로세서 형태로 구성함으로써 효율적인 구조를 제안하였다. 즉, Telephony Board Control 모듈이 메인 프로그램이 되어 시스템 구성시 보드 종류와 무관하게 동작될 수 있도록 미들웨어를 구현함으로써 하드웨어의 교체에 따른 수고를 줄이게 되며 시스템의 확장성 및 독립성을 가지게 된다.

본 논문에서는 VoiceXML 표준에 따라서 작동되어지는 IVR 서버를 실제로 설계하고 구성하는데 있어서 향후 서비스의 변경 및 확장에 대비하여 어떻게 구현하는 것이 효과적인가를 고려하여 작성하였다. 그러므로 이 논문의 구현된 시스템이 기존의 IVR 서버에 비하여 성능을 향상시켰다고 말할 수는 없다. 시스템의 성능 향상 문제는 주로 실시간 끝점검출 알고리즘의 성능향상, 음성인식기와 음성합성기 그리고 Telephony Board Control 부분의 프로그램 능력에 따라 결정된다고 볼 수 있다. 본 논문에서 시스템의 성능 개선보다는 시스템 구성시에 IVR 서버 설계자가 향후 시스템의 확장 및 변경시에 좀더 적은 노력으로 서버를 구축할 수 있도록 구조를 제안하여 성능을 점진한 것에 주안점을 두었다.

향후 연구되어야 할 과제로는 VoiceXML 인터프리터의 성능에 따른 신뢰성 있는 IVR 서버를 구현하기 위해 여러 가지 보완 루틴을 Telephony Board Control 부분에 추가할 필요가 있다. 예를 들자면, 실시간 음성검출을 위한 실시간 끝점 처리 알고리즘을 사용하여 Telephony Board에서 지원하는 음성 검출을 위한 기능을 보완하도록 하는 것을 들 수 있다. 또한 VoiceXML 인

터프리터의 해석 능력에 따라서 VoiceXML 실행부와 인터프리터의 역할을 구분하는 명확한 정책을 세워서 시스템을 구성할 필요도 있다. 마지막으로 본 논문에서는 VoiceXML Interpreter와 Telephony Board Control간의 정보전달을 위한 방법으로 INAP 형식을 사용하여 IVR 서버를 구축하였으나, INAP 형식과 함께 CTI의 표준인 S.100, S.200, S.300 등의 표준안을 근거로 한 IVR 서버 구축을 진행할 계획이다.

참 고 문 헌

- [1] <<http://www.voicexml.org>>
- [2] Ball, Thmas, Veta Bonnewell, Peter Danielsen, Peter Mataga & Kenneth Rahor. 2000. "Speech-Enabled Services Using TelePoral Software and VoiceXML." *Bell Lab Technical Journal*.
- [3] DRAFT NEW ITU-T RECOMMENDATION Q.1228
- [4] Enterprise Computer Telephony Forum, <<http://www.ectf.org>>
- [5] 강선미. "VoiceXML 한국어 인터프리터 개발을 위한 연구." 정보통신학술연구과제 자유연구 00-23.

접수일자: 2002. 7. 25.

게재결정: 2002. 8. 23.

▲ 이창호

서울특별시 성북구 안암동 5가 고려대학교 산학관 6층 (우: 136-701)
 보이스미디어텍(주)
 Tel: +82-2-923-8830 Fax: +82-2-923-8830
 E-mail: chlee@voicemediatech.com

▲ 장원조

서울특별시 성북구 정릉동 16-1 (우: 136-704)
 서경대학교 컴퓨터학과
 Tel: +82-2-940-7291 Fax: +82-2-919-5075
 E-mail: wjjang@ihci.skuniv.ac.kr

▲ 강선미

서울특별시 성북구 정릉동 16-1 (우: 136-704)
 서경대학교 컴퓨터학과
 Tel: +82-940-7291 Fax: +82-2-919-5075
 E-mail: smkang@skuniv.ac.kr