# 사용자 주도 폼 다이얼로그 시스템의 VoiceXML 어플리케이션에 관한 연구

권 형 준[†] · 노 용 완[††] · 이 현 구[†††] · 홍 광 석[††††]


## 요 약

VoiceXML은 음성을 통해 웹 자원 탐색을 제공하기 위한 목적으로 설계된 XML 기반의 새로운 마크업 언어이다. VoiceXML로 만들어진 어플리케이션은 기계 주도 폼 다이얼로그 구조와 상호 주도 폼 다이얼로그 구조로 분류된다. 이와 같은 다이얼로그 구조들은 어플리케이션 개발자에 의해 서비스 시나리오가 결정되기 때문에 사용자가 자유롭게 웹 자원을 탐색하는 서비스를 구축할 수 없다. 본 논문에서는 사용자의 의도에 따라 서비스 시나리오가 결정되는 음성 웹 서비스의 구축을 위해 사용자 주도 폼 다이얼로그 시스템의 VoiceXML 어플리케이션 구조를 제안한다. 제안하는 어플리케이션은 사용자에 의해 요청된 정보로부터 인식 후보들을 자동적으로 검출하여 음성 앵커로 사용하고 각각의 음성 앵커를 새로운 음성 노드로 연결한다. 제안하는 시스템의 예로 IT 용어사전을 내장한 뉴스 서비스를 구현하여 음성 앵커의 검출 및 등록 여부를 확인하였고, 음성 인식률 및 사용자가 의도한 정보를 성공적으로 제공했는지 판단하는 척도가 되는 적중률과 응답 속도를 측정하였다. 실험 결과, 제안한 시스템이 기존의 VoiceXML 폼 다이얼로그 구조의 시스템보다 더 자유로운 웹 자원의 탐색이 가능함을 확인하였다.


**키워드 : 보이스엑스엠엘, 다이얼로그, 앵커, 노드**

# A Study on VoiceXML Application of User-Controlled Form Dialog System

Hyeong-Joon Kwon[†] · Yong-Wan Roh[††] · Hyon-Gu Lee[†††] · Kwang-Seok Hong[††††]


## ABSTRACT

VoiceXML is new markup language which is designed for web resource navigation via voice based on XML. An application using VoiceXML is classified into mutual-controlled and machine-controlled form dialog structure. Such dialog structures can't construct service which provide free navigation of web resource by user because a scenario is decided by application developer. In this paper, we propose VoiceXML application structure using user-controlled form dialog system which decide service scenario according to user's intention. The proposed application automatically detects recognition candidates from requested information by user, and then system uses recognition candidate as voice-anchor. Also, system connects each voice-anchor with new voice-node. An example of proposed system, we implement news service with IT term dictionary, and we confirm detection and registration of voice-anchor and make an estimate of hit rate about measurement of an successive offer from information according to user's intention and response speed. As the experiment result, we confirmed possibility which is more freely navigation of web resource than existing VoiceXML form dialog systems.


**Key Words : VoiceXML, Dialog, Anchor, Node**

## 1. 서 론

IBM과 모토로라 및 AT&T 등의 세계적인 기업들은 VoiceXML 포럼을 개설하고 웹 자원의 접근 수단에 있어서 시공간의 제약을 완화할 수 있도록 보이스와 오디오를 이용한 접근 방법을 연구하였다. 그 결과, 웹의 표준을 제정하는 W3C는 VoiceXML을 국제 표준으로 승인하여 현재 표준인 VoiceXML 2.0의 골격이 되는 1.0이 등장하였다. 유무선 전화를 이용해 웹 자원에 접근하고 탐색할 수 있는 인터페이스를 제공하는 VoiceXML은 음성 서비스의 제공을 위해 값비싼 하드웨어 비용이나 난해한 개발 언어 환경에서 벗어나게 해 줌으로서 일반 개발자도 손쉽게 음성 인식 및 합성을 이용한 서비스를 개발할 수 있게 한다[1].

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. IITA-2006-(C1090-0603-0046)
† 준 회 원 : 성균관대학교 대학원 정보통신공학부 석사과정(교신저자)
†† 준 회 원 : 성균관대학교 대학원 정보통신공학부 박사과정
††† 정 회 원 : 서일대학 정보기술계열 정보통신전공 교수
†††† 종신회원 : 성균관대학교 정보통신공학부 교수
논문접수 : 2006년 11월 22일, 심사완료 : 2007년 5월 4일

컴퓨터의 경이로운 진보와 이에 동반한 가격의 안정화는 인터넷을 이용하는 사람들이 컴퓨터를 이용한 비주얼 환경의 웹 브라우징을 가능하게 하였으며, VoiceXML의 등장으로 텔레포니와 음성기술을 결합할 수 있게 되었다. 현재는 원격 통신과 웹이 수렴하는 지점에 서 있으며 미래에는 보이스 환경에서의 웹 브라우징을 위한 인터페이스의 중추에 VoiceXML이 자리할 것이라 예상할 수 있다[2].

VoiceXML 어플리케이션으로 구축된 서비스의 동작 형태는 사용자가 시스템의 질문에 음성 및 DTMF(Dual Tone Multi Frequency)로 응답하는 모습이다. 시스템은 사용자의 응답을 데이터로 하여 음성 합성으로 적절한 콘텐츠를 제공하는데, 시스템은 사용자의 응답에 따라 시나리오의 진행에 약간의 변화를 주기도 한다. 폼 구조로 동작하는 VoiceXML에서는 이러한 형태의 서비스들을 제공하기 위한 어플리케이션의 구조를 기계 주도 폼 다이얼로그와 상호 주도 폼 다이얼로그로 구분한다[3]. 기계의 질문과 사용자의 응답이 반복되는 기계 주도 폼에 비해 상호 주도 폼은 사용자의 응답에 맞게 시나리오가 분기되어 보다 다양한 형태의 콘텐츠 제공이 가능하다. 이러한 기존의 폼 구조들의 한계를 극복하기 위한 연구 결과들로 서버 측면의 스크립트 언어를 이용하여 상호 주도 폼을 강화한 방법과 시나리오 진행 상황을 비주얼 환경으로 제공함과 아울러 사용자의 제어기능을 추가한 시스템 및 XML의 문서 변환기능을 이용하여 보이스 및 비주얼 환경을 동시 제공하는 멀티모달 시스템 등이 있다[4-6]. 하지만 결국 시스템이 주도하는 시작과 끝이 하나인 획일적인 시나리오를 갖는 점은 동일하기 때문에 적용할 수 있는 서비스가 제한적이어서 인건비 절감을 위해 ARS의 대용으로 사용되고 있는 것이 현실이다.

본 논문에서는 사용자의 주도로 시나리오를 만들어가는 사용자 주도 폼 다이얼로그 시스템을 제안한다. 제안하는 시스템은 비주얼 환경에서 웹 자원을 탐색하는 것과 유사한 방법으로 시나리오를 만들어 가기 때문에 기존의 음성을 통한 콘텐츠 제공 및 자원 탐색의 한계를 극복할 수 있어서 기존의 획일화된 서비스와 다른 새로운 형태의 서비스를 제공할 수 있다. 이러한 시스템의 설계 및 구현을 위해 하이퍼텍스트 및 하이퍼미디어의 앵커와 노드 구조를 적용하고, 개발자의 개입이 없이 사용자가 요구하는 정보에 기반을 둔 음성 앵커와 음성 노드의 자동 등록 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서 기본적인 VoiceXML 폼 다이얼로그 구조와 그것의 근간이 되는 폼 해석 알고리즘을 설명한다. 3장에서는 기본적인 VoiceXML 폼 다이얼로그 구조의 문제점 및 한계와 이를 해결하기 위한 기존 연구를 설명하고, 4장에서는 제안하는 시스템의 구조 및 특징과 구현을 위한 주요 알고리즘들을 소개한다. 5장에서는 제안하는 시스템의 검증을 위한 실험 및 그 결과를 도출하고, 6장에서 결론을 맺는다.

## 2. VoiceXML 다이얼로그

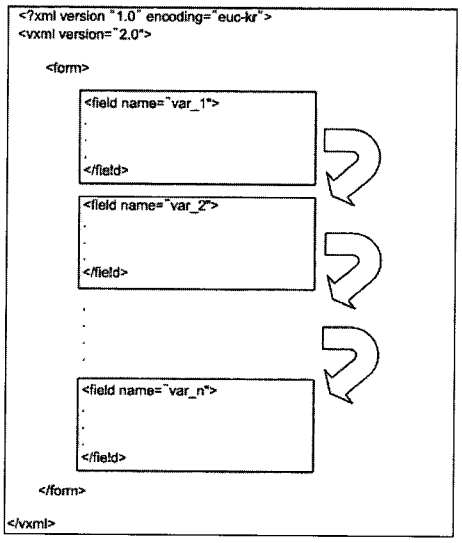
### 2.1 Form Interpretation Algorithm

Form Interpretation Algorithm(FIA)는 VoiceXML 문서에 있는 폼과 폼 아이템의 실행 순서를 결정하는데, 폼 컨텍스트에 해석기가 진입했을 때 다음과 같은 4단계가 순차적으로 실행된다. 첫째는 초기화 단계로서 이전에 사용자와 소통한 과정에서 만들어진 폼의 기록을 전부 없애는 역할을 수행한다. 둘째는 현재의 폼에서 다음으로 실행할 폼 요소를 선택하는 선택 단계이며, 셋째는 수집 단계로서 사용자에게 출력할 프롬프트를 선정하는 것으로 시작한다. 프롬프트가 출력되면 사용자의 음성 혹은 DTMF신호를 수집한다. 네 번째 단계인 처리 단계에서는 수집한 데이터를 바탕으로 사용자에게 적절한 응답을 하거나 흐름을 되돌리는 등의 역할을 수행한다. 각 단계에서는 해당 단계가 실행될 필요조건이 충족되지 않으면 실행되지 않고 건너뛰어 다음 단계가 실행되며 네 번째 단계의 실행이 끝난 후에 새로운 문서로 이동하지 않을 경우는 두 번째 단계로 이동한다[1].

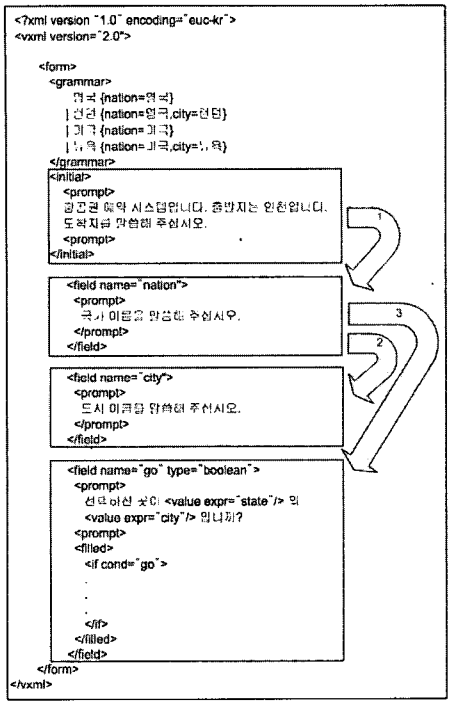
서버 측면의 스크립트 언어를 사용하지 않는 VoiceXML 어플리케이션은 전적으로 FIA가 폼의 실행 순서를 결정한다. 이러한 어플리케이션은 서비스 제공에 있어서 유연성이 떨어지지만 프로그램상의 오류나 서버의 오동작을 최소한으로 줄일 수 있다.

### 2.2 기계 주도 폼 다이얼로그

VoiceXML로 구축된 서비스를 이용할 때는 인간과 인간의 의사소통에 있어서 가장 효과적인 수단인 음성으로 컴퓨터와 대화하는 방식을 취한다[2]. 그러나 실제로 대화가 이루어지는 형태를 살펴보면 컴퓨터는 일방적으로 필요한 질문을 하고 사용자는 이에 대한 대답만을 해야 한다. 이러한 대화 형태를 VoiceXML에서는 기계 주도 폼 다이얼로그라 부른다. 즉, VoiceXML 문서에 <field> 태그가 기술된 순서로 사용자의 입력을 받아들이는 다이얼로그는 기계 주도 폼 다이얼로그라고 할 수 있다. (그림 1)은 사용자의 입력을 n번 받아들이는 전형적인 기계 주도 폼 다이얼로그의 모습을 나타낸다[3].




(그림 1) 기계 주도 폼 다이얼로그의 실행 순서




(그림 2) 상호 주도 폼 다이얼로그의 실행 순서

### 2.3 상호 주도 폼 다이얼로그

상호 주도 폼 다이얼로그는 하나의 폼에 여러 개의 필드가 있을 때 실행되는 필드의 순서가 스크립트에 기술된 순서대로 실행되는 기계 주도 폼 다이얼로그와는 달리, 사용자의 입력에 따라 순서가 바뀌는 다이얼로그 시스템이다.

(그림 2)는 상호 주도 폼 다이얼로그 시스템의 실행의 설명을 위한 예이다. <grammar> 에는 사용자에 의해 입력될 수 있는 인식 후보들이 나열된다. 해당 인식 후보를 발성했을 때 각각 괄호 안의 내용대로 변수에 값이 할당된다. 할당된 값에 따라서 요소의 실행 순서가 결정되는데, 그림 2의 경우는 1번 순서로 실행된 후 사용자의 입력에 따라서 2번 순서로 실행할지, 2번 순서를 건너뛰고 3번 순서로 진행할지의 여부를 판단한다. 설명한 것과 같이 상호 주도 폼은 사용자의 의사에 따라 프로그램의 진행에 약간의 영향을 줄 수 있다[3].

## 3. 기존 폼 다이얼로그의 문제점 및 VoiceXML의 한계

컴퓨터와 텔레포니의 통합 및 음성을 통한 인터페이스의 제공을 용이하게 만들어 주는 점에서 VoiceXML은 큰 역할을 한다. 그럼에도 불구하고, 현재 VoiceXML 기반의 시스템은 몇 가지 문제점과 VoiceXML의 범위 안에서 극복할 수 없는 한계들을 가지고 있다. 첫째로, VoiceXML은 개발자가 정의한 <grammar>, 즉 인식 후보에 정의되지 않은 단어는 인식하지 못한다[1]. 개발자가 사용자가 입력할 가능성이 있는 모든 단어를 인식 후보에 정의하면 이 단점을 어느 정도 해결할 수 있지만 너무 많은 단어를 인식 후보에 등록하면 사용자의 입력을 정확히 인식하지 못한 경우 사용자가 요청한 정보가 아닌 다른 정보를 출력할 가능성이 높아지기 때문에 의도하지 않은 시나리오 진행이 이루어질 수 있다. 이와 같은 문제의 해결을 위해 멀티모달 인식 시스템이 등장하였다. 비주얼과 보이스를 병행하는 멀티모달 인식 시스템은 인식 후보를 사용자가 눈으로 확인할 수 있으므로 개발자가 정의한 인식 후보를 보며 사용자가 음성으로 입력할 수 있는 장점이 있다. 둘째, VoiceXML의 현재 폼 다이얼로그 구조로는 전적으로 개발자의 의도에 따라 프로그램이 진행되므로 사용자의 입장에서 자유롭게 서비스를 이용하는 것이 불가능하다. VoiceXML 어플리케이션의 설계에 있어서 개발자의 의도에 의존하는 시나리오 흐름의 보완을 위한 기존의 연구에서 음성 앵커의 개념이 제안된 바 있다. 불필요한 정보를 건너뛰기 위한 "*Skip*", 정보를 다시 청취하기 위한 "*Repeat*", 지나간 정보를 다시 청취하기 위한 "*Back*", 시나리오의 처음부터 다시 실행하기 위한 "*Begin*", 프로그램의 이용을 정지하기 위한 "*Stop*" 등의 정적인 음성 앵커들을 강제로 인식 후보에 추가한 상대적 방향 제어가 그 예이다[5]. 이와 같은 기존의 연구는 기계 주도 폼과 상호 주도 폼을 보조하여 보다 유연한 음성 어플리케이션의 구현을 가능하게 한다. 그렇다고 하더라도, 폼 다이얼로그의 기본 구조 및 그 시나리오의 흐름에서 크게 벗어났다고 보기 어렵다. 셋째, 현재 VoiceXML의 기본 골격인 폼 다이얼로그 구조는 사용자가 입력한 내용을 인식하지 못해서 오류 메시지를 출력하게 되면 사용자의 입력을 받기 위한 음성 합성 안내 메시지를 다시 듣게 되는 상황이 발생한다. VoiceXML의 현재 표준으로는 본 문제를 해결할 수 없다[5]. 기타 문제점으로는 VoiceXML의 음성 인식률과 음성 합성음의 억양 등의 문제를 거론할 수 있다. 이러한 문제점은 VoiceXML이 음성 입력을 인식하고 출력을 위해 합성할 때에 ASR과 TTS 엔진에 전적으로 의존하기 때문에 VoiceXML 범위 안에서의 해결이 불가능하다[5].

이와 같은 VoiceXML의 문제점 및 한계들은 보이스 환경의 웹 자원 탐색을 지향하는 원래의 목적과 동떨어진 서비스를 제공하게 만드는 결과를 낳았다. 간단한 예로 텔레포니 시스템 구축의 비용 절감을 위한 ARS의 대용으로 사용되거나, 상담원 인건비 절감을 위한 콜 센터의 구축에 사용되고 있다. 물론 보이스 포탈이라는 명칭을 가진 서비스가 있기는 하지만 웹 브라우저를 이용한 비주얼 환경에서의 웹 자원 탐색과 비교했을 때 실제 포탈로서의 역할을 전혀 수행하지 못하고 있다. 이에 따라, 본 논문에서는 이와 같은 문제점과 현재 VoiceXML의 활용도가 그 기능에 비해 미약함을 인지하고 비주얼 환경의 웹 자원 탐색과 유사한 구조 및 형태를 가지는 사용자 주도 폼 다이얼로그 시스템을 제안한다.

## 4. 사용자 주도 폼 다이얼로그 시스템

제안하는 시스템은 앞 장에서 설명한 VoiceXML의 문제점 중에 두 번째 문제점인 획일화된 시나리오를 가지는 VoiceXML의 폼 다이얼로그 구조에 관한 것이다. 본 논문에서는 서버 측면 스크립트의 도움을 받아 기존의 개발자 주

도적인 시나리오를 벗어나 사용자의 입장에서 보다 자유로운 서비스의 이용이 가능한 시스템의 설계와 구현 방법을 제시한다. 제안하는 시스템은 마치 비주얼 환경의 웹 브라우징과 유사한 형태의 자원 탐색 기법을 사용한다. 제안하는 시스템인 사용자 주도 폼 다이얼로그 구조의 VoiceXML 어플리케이션은 VoiceXML의 기본적 다이얼로그 구조인 기계 주도 폼과 상호 주도 폼은 물론, 이를 극복하기 위해 어플리케이션 개발자가 정적인 앵커를 추가했던 기존의 연구보다 더 사용자 지향적인 시스템이다. 사용자가 요청한 정보로부터 음성 앵커를 자동으로 추출하고 인식 후보에 등록하여 새로운 음성 노드들과 연결시켜 사용자로 하여금 원하는 시나리오를 만들어가며 서비스를 이용할 수 있게 만든다. 본 장에서는 제안하는 시스템의 기본 구조와 구현 방법을 설명한다.

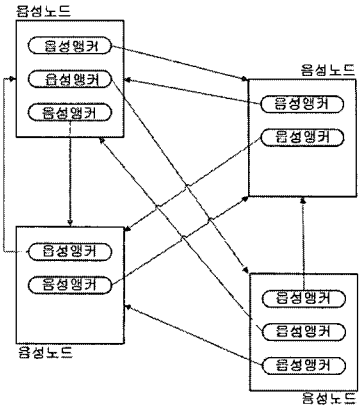
### 4.1 하이퍼미디어와 음성 앵커 및 음성 노드

하이퍼미디어의 기원은 하이퍼텍스트로부터 시작된다. 일반 텍스트는 일정한 정보를 순차적으로 얻게 되지만 하이퍼텍스트는 사용자가 연상하거나 의도하는 순서에 따라 원하는 정보를 얻을 수 있는 시스템이다. 텍스트간의 연결이 아닌 텍스트와 기타 미디어(그림, 사진, 음성, 영상 등)가 연결되는 것을 하이퍼미디어라 칭하며, 하이퍼미디어 및 하이퍼텍스트에서는 연결되는 정보들의 시작점을 앵커라 하고 연결 대상을 노드라 한다. 본 논문에서는 사용자 주도 폼 다이얼로그 구조의 음성 어플리케이션을 위해서 하이퍼텍스트 및 하이퍼미디어를 이루는 개념인 앵커와 노드 구조를 VoiceXML 어플리케이션에 적용하고 음성 앵커와 음성 노드로 사용한다. (그림 3)에 제안하는 시스템의 VoiceXML 문서 구조를 간략하게 표현하였다.

제안하는 사용자 주도 폼 다이얼로그 시스템은 하나의 VoiceXML 문서에 하나의 폼이 존재한다. 정보를 제공하기 전에 제공될 정보의 내용을 처리하여 인식 후보를 추출해 등록하고 폼을 생성하여 음성 앵커로 사용한다. 음성 앵커는 다른 정보와 연결되며, 연결되는 정보에는 또 다른 잠재적인 음성 앵커들이 있어서 또 다른 정보로 연결된다. 이와 같은 구조는 비주얼 환경의 웹 탐색 방법과 흡사하여 기존에는 제공할 수 없었던 서비스의 제공을 가능하게 하고 사용자의 의도에 따라 원하는 정보를 탐색해가며 시나리오를 만들어 갈 수 있게 한다.

(그림 2)의 VoiceXML 문서에서 나타나는 것처럼 음성 앵커는 곧 인식 후보이며 <grammar> 태그 안에 정의된다. 본 논문에서는 <grammar> 태그 안에 정의할 음성 앵커를 서버 측면의 스크립트 언어를 사용해 동적으로 정의한다.

그 음성 앵커들은 사용자가 요구한 정보로부터 추출되고, 음성 앵커가 정의됨과 동시에 각각의 음성 앵커와 연결되는 새로운 음성 노드가 생성된다. 사용자가 음성 앵커를 입력하면 그와 연결된 음성 노드로 이동하고, 또 새로운 음성 앵커를 검출해 낼 것이다. 실제로 제안하는 시스템에서 기본적으로 음성 노드를 생성할 수 없는 경우에는 음성 앵커가 정의되지 않는다. 이는 개발자의 의도에 전적으로 의존하던 기존의 VoiceXML 어플리케이션과 비교했을 때 사



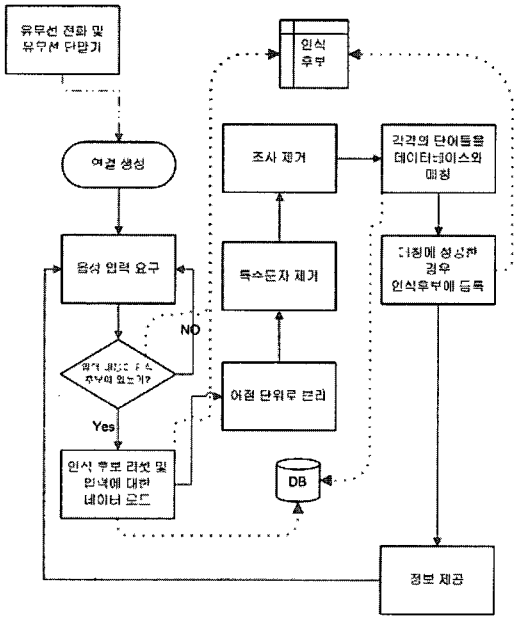

(그림 3) 음성 앵커 및 음성 노드간의 관계

용자 입장에서 더 자유로운 정보 검색을 가능하게 만들고, 기존에 제공할 수 없었던 새로운 유형의 서비스를 제공할 수 있게 해 준다.

### 4.2 실행 과정

(그림 4)는 제안하는 다이얼로그 시스템의 실행 과정을 나타낸다.

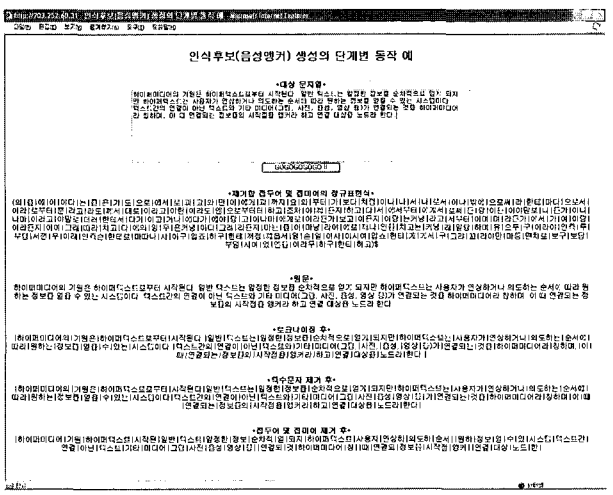
연결 요청이 이루어지고 연결이 생성되면 사용자의 음성 입력을 요구한다. 사용자의 음성 입력에 따라 사용자의 다음 입력에 대한 인식 후보가 갱신된다. 인식 후보는 사용자에게 제공될 내용과 미리 준비한 데이터베이스에 기록되어 있는 데이터를 기반으로 동적으로 등록된다. 따라서 기존의 기계 주도 폼과 상호 주도 폼의 시나리오 진행 구조와는 다르게 제안하는 어플리케이션의 시나리오는 (그림 3)과 같은 관계들의 집합으로 만들어지며 마치 그물과 흡사한 구조를 갖게 될 것이다.




(그림 4) 사용자 주도 폼 다이얼로그 구조의 VoiceXML 어플리케이션 실행 순서도

### 4.3 인식 후보 수집

제안하는 VoiceXML 어플리케이션의 구현을 위해서는 제공될 정보에 기반하는 동적인 인식 후보의 수집이 필요하다. 동적인 인식 후보의 수집을 위해서 ASP 및 JSP, PHP 등의 서버 측면의 스크립트 언어를 활용한다. 수집된 인식 후보들이 곧 음성 앵커이며, 음성 앵커와 연결되는 새로운 정보는 음성 노드이다. 인식 후보 수집의 절차는 총 4단계로 분류하는데, 첫째로 VoiceXML의 음성 인식은 기본적으로 고립단어인식을 사용하므로 사용자에게 제공할 정보에 담긴 단어의 리스트를 만들기 위해 제공될 정보를 문장의 공백을 기준으로 나누어 어절 단위로 분리한다. 어플리케이션을 개발하기 위한 프로그램 언어들은 특정 문자를 기준으로 문자열을 나누는 API를 제공하므로 해당 언어의 기술 문서 혹은 설명서를 참고한다. 둘째, 첫 번째 단계에서 분리한 각 어절들을 이루는 문자들을 검사하여 특수문자인 경우 그 문자를 삭제한다. 특수문자를 가리키는 아스키 코드들과 비교하여 공백으로 대체한 후 공백을 삭제하는 것이 가장 효율적인 방법이다. 첫 번째 문자부터 비교를 반복하여 마지막 문자의 비교를 마치면 전체 문자열에 포함된 공백을 모두 삭제한다. 셋째, 특수문자가 제거된 어절에서 조사를 정규표현식(Regular Expression)을 활용하여 제거한다. 정규표현식이란, 특별한 기호들을 이용하여 만들어진 일종의 패턴으로서 일련의 데이터를 가리킬 수 있는 표현식이다[7]. 정규표현식으로 작성된 패턴과 특수문자가 제거된 어절들의 문자열 형식을 비교하여 패턴에 부합하는 문자열 구조일 경우 조사를 제거한다. 이 때, 제거할 조사 목록을 제공할 서비스에 적합하게 작성할 필요가 있다. 넷째, 특수문자와 조사가 제거된 어절들을 미리 구축해 둔 데이터베이스의 키워드 항목과 비교한다. 세 번째 단계까지 처리된 데이터들 중 데이터베이스의 키워드 항목에 기록되어 있는 요소들이 인식 후보로 등록되어 음성 앵커로 사용된다. 인식 후보를 수집하기 위해 활용하는 데이터베이스 테이블에 필수적으로 포함되어야 할 최소한의 항목들은 고유 번호 형식의 식별자, 내용을 대표하는 키워드, 사용자에게 제공될 정보이다.




(그림 5) 인식 후보 생성의 단계별 동작 예

(그림 5)는 인식 후보가 수집되는 단계들의 중간 과정을 살펴보기 위한 화면이다. IIS를 기반으로 Active Server Page를 이용하여 작성된 VoiceXML 문서를 XML 해석기를 내장한 Internet Explorer 6 SP2로 출력하였다. 입력 문장에서 조사들이 제거되어 단어만 남았음을 알 수 있다. 영문의 경우는 조사를 제거하는 과정이 없어도 꽤 효과적인 음성 앵커 추출이 가능하다.

### 4.4 주요 알고리즘

앞서 설명한 네 단계에서 첫 번째 단계인 어절 단위로 분리하는 방법과 세 번째 단계 중 일부인 조사 제거를 위한 방법을 알고리즘 1에 기술하였다. 알고리즘의 소개에 활용한 소스코드는 Active Server Page가 지원하는 정규표현식을 활용하여 작성되었다. 본 소스코드를 실제 적용할 때에는 제거할 조사에 맞도록 정규표현식을 수정하여야 하며, 정규표현식은 알고리즘 1의 4번째 줄에 정의한 Patrn 변수에 지정한다. 알고리즘 1은 '의', '을', '에' 의 3가지 조사만을 제거한다.

앞서 설명한 네 단계의 절차로서 처리된 결과 문자열들과 데이터베이스에 기록된 키워드를 비교하는 효과적인 방법을 알고리즘 2에 기술하였다. 알고리즘 2는 데이터베이스에 질의하여 얻어온 키워드들을 특정 배열에 저장했다고 가정하고 작성되었다. 곧, 두 배열 요소들의 1:1 비교 알고리즘이라 할 수 있다. GoF가 정의한 디자인 패턴에서는 이 알고리즘을 무엇인가가 많이 모여 있는 것 중에서 하나씩 끄집어내어 열거하며 전체를 검색하는 처리 방법을 가리키며 Iterator 패턴이라 칭하고 있다[8]. 알고리즘 2를 객체지향 기반의 언어로 구현할 때에는, 유지와 보수 측면을 고려하여 Iterator 패턴을 일반화한 클래스 및 인터페이스를 만들어 사용함이 바람직하다.

(알고리즘 1) 문자열 토크나이징 및 정규표현식 활용 예

```
Option Explicit
Dim TestString="나는 학교에 간다." '처리할 문자열
Dim TokenString=Split(TestString) '공백 단위 토크나이징
Dim Patrn="(의|을|에|)$" '정규표현식

'조사 제거 결과를 배열에 저장
For i=0 to Ubound(TokenString)
  TokenString(i)= Trim(RegExpReplace(Patrn,TokenString(i), " "))
Next

Public Function RegExpReplace(Patrn,TrgtStr,RplcStr)
  Dim ObjRegExp
  On Error Resume Next
  Set ObjRegExp=New RegExp
  ObjRegExp.Pattern=Patrn
  ObjRegExp.Global=True
  ObjRegExp.IgnoreCase=True
  RegExpReplace=ObjRegExp.Replace(TrgtStr,RplcStr)
  Set ObjRegExp=Nothing
End Function
```

(알고리즘 2) Iterator 패턴

```
Begin
  For i=0 to Array1_size
    For j=0 to Array2_size
      If Array1_element[i] = Array2_element[j] then
        Print "Same"
      Else
        Print "Different"
      End if
    Next
  Next
End
```

앞의 두 알고리즘을 이용하여 사용자에 의해 요구된 정보로부터 음성 앵커를 추출하고 다음번의 입력에서 사용될 인식 후보에 등록한다. 이 작업은 사용자가 음성 입력을 시도할 때 마다 반복된다. 이는 획일화된 기존의 VoiceXML 어플리케이션 시나리오에서는 찾아볼 수 없었던 본 연구의 새로운 특징이다. 개발자에 의해 시나리오가 정해졌던 기존의 어플리케이션과는 달리 사용자가 시나리오를 만들어 나간다. 즉, 기존의 전통적인 VoiceXML 어플리케이션에서 사용되었던 시나리오의 개념이 희미해지고 VoiceXML의 원래 목적인 음성을 이용한 웹 서핑을 가능하게끔 만든다.

## 5. 실험 및 결과

제안하는 시스템의 구현 및 실험을 통한 확인을 위해서 IT 용어사전을 내장한 뉴스 서비스를 구현하였다. 본 서비스는 뉴스 기사의 제목을 안내하고 사용자가 선택한 뉴스의 본문을 음성으로 들려주는 서비스이다. 뉴스 본문이 사용자에게 전달되는 중에 사용자는 IT 용어를 음성으로 입력할 수 있으며, 시스템은 그에 대한 응답으로 입력된 IT 용어에 대해 설명한다. 음성 앵커 검출을 위한 정규표현식의 작성은 국립국어원에서 공개한 현대 국어 사용 빈도 조사표에서 조사 목록을 참고하였다[9].

### 5.1 동작 환경

제안하는 시스템의 음성인식 및 합성을 위한 ASR 및 TTS 엔진과 VoiceXML 해석기는 (주)KT의 휴보이스 솔루션을 사용하였다. 전화와 컴퓨터의 상호작용을 위한 CTI 보드는 동시에 12 채널을 수용할 수 있는 Intel Dialogic D/120JCT-LS를 장착하였다. <표 1>은 제안한 서비스의 실험을 위해 구성한 서버의 하드웨어 및 소프트웨어의 전반적인 사양을 나타낸다.

제안하는 시스템의 구현을 위해 사용한 데이터베이스 테이블은 총 2개이다. 뉴스 기사가 저장되는 첫 번째 테이블은 고유번호, 기사의 제목, 기사의 본문, 기사가 기록된 시간을 담고 있으며, 두 번째 테이블은 고유번호, 용어, 용어의 뜻, 접근 횟수를 항목으로 가지고 있다.

<표 1> 실험에 이용한 서버 사양

| 항목 | 설명 |
|---|---|
| CPU | Intel Xeon 3.0Mhz |
| RAM | 2GB |
| CTI Board | Intel Dialogic D/120JCT-LS |
| OS | Windows 2000 Professional SP4 |
| IIS Ver. | 5.0 |
| RDBMS | MS-SQL 6.0 |
| ASR | KT Huvois |
| TTS | KT Huvois |

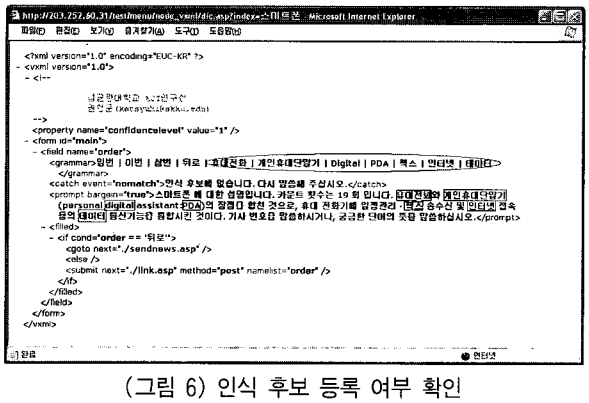
### 5.2 실험 방법 및 결과

제안하는 시스템의 실험을 위해 XML 해석기를 통하여 비주얼 환경에서 인식 후보의 동적 등록 여부를 확인한다. 최대 7글자 이하의 한국어 고립단어를 사용하여 서비스의 원활한 제공을 판단하기 위한 척도로서 인식 성공률과 응답 속도를 측정한다. 마지막으로, 제안하는 어플리케이션의 목적을 달성했느냐에 대한 판단, 즉 사용자가 얼마나 자유로운 시나리오를 진행했는지에 대한 객관적 근거로 사용자가 발성한 키워드의 적중률을 측정한다.

#### 5.2.1 인식 후보 등록

제안한 시스템이 인식 후보를 얼마나 검출하여 등록할 수 있는지 확인하기 위해 XML 해석기를 내장한 Internet Explorer 6 SP2를 이용해 VoiceXML 문서에 접근하고 그 결과를 (그림 6)에 나타냈다. 본 실험 결과로서 보이는 (그림 6)에서 사용자에게 제공될 정보가 포함하고 있는 IT용어들이 인식 후보에 등록되어 음성 앵커로 사용됨을 확인할 수 있다.

(그림 6)은 사용자가 "스마트폰" 을 발성하여 그에 관한 정보를 요청했을 때 생성되는 VoiceXML 문서를 비주얼 화면으로 출력한 것이다. <prompt> 태그가 가지고 있는 문자열이 사용자가 TTS를 통해 청취하게 되는 정보이며, 청취한 후에는 <grammar> 태그가 가지고 있는 인식 후보 중 한 가지를 발성하여 새로운 정보를 요구할 수 있다. 4장에서 소개한 알고리즘들을 통해서 (그림 6)에 파란색으로 표기된 단어들이 <grammar> 안에 자동으로 등록되어 음성 앵커로 활용됨을 확인할 수 있다. 4장에서 설명한 것과 같이 얼마나 많은 단어를 검출하여 음성 앵커로 등록할 수 있는지에 대한 관건은 데이터베이스가 가진 키워드 및 그에 따른 정보량에 달려 있다.



(그림 6) 인식 후보 등록 여부 확인

5.2.2 인식 성공률 및 응답 속도

제안하는 시스템은 제공할 콘텐츠에 따라서 방대한 문자열 데이터의 처리가 요구될 수 있으며, 새로운 정보를 제공할 때마다 데이터베이스에 기록된 내용을 기반으로 동적인 VoiceXML 문서를 생성한다. 실제 서비스 제공에 있어서 장애 여부를 판단하기 위해 음성 서비스 경험자 및 무경험자를 각각 5명씩 총 10명의 피험자를 대상으로 일상 소음이 존재하는 사무실에서 피험자들이 본 어플리케이션을 사용하면서 발성한 단어를 가지고 인식 성공률과 응답 속도를 측정하여 <표 2>에 나타냈다.

〈표 2〉 인식 성공률 및 응답 속도 측정 결과

| 피험자 구분 | 피험자 | 인식 성공 횟수 | 인식률 (%) | 평균 응답 속도 |
|---|---|---|---|---|
| 경험자 | A1 | 9 / 10 | 90 | 최대 1.3초 |
| | A2 | 9 / 10 | 90 | 최대 1.0초 |
| | A3 | 9 / 10 | 90 | 최대 1.2초 |
| | A4 | 10 / 10 | 100 | 최대 1.2초 |
| | A5 | 9 / 10 | 90 | 최대 1.0초 |
| 경험자 평균 | | 47 / 50 | 92 | 최대 1.14초 |
| 무경험자 | B1 | 8 / 10 | 80 | 최대 1.0초 |
| | B2 | 6 / 10 | 60 | 최대 1.2초 |
| | B3 | 7 / 10 | 70 | 최대 1.1초 |
| | B4 | 6 / 10 | 60 | 최대 1.2초 |
| | B5 | 6 / 10 | 60 | 최대 1.1초 |
| 무경험자 평균 | | 33 / 50 | 66 | 최대 1.16초 |

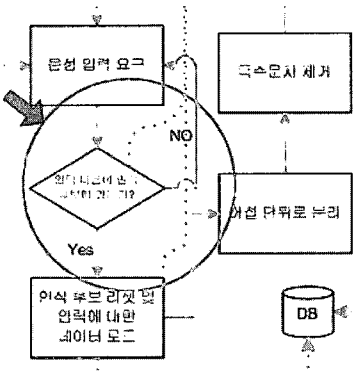
음성 인식률 측정 결과를 살펴보면, 음성 서비스 경험자는 92%의 음성 인식률을 보였고 음성 서비스 무경험자는 66%의 음성 인식률을 보였다. 제안하는 구조를 적용하지 않은 기존의 VoiceXML 어플리케이션의 음성 서비스 경험자와 무경험자의 평균 인식률이 각각 86%와 72%였음에 근거했을 때, 음성 서비스를 경험하지 못했던 사용자는 우리가 제안하는 새로운 구조의 어플리케이션에 적응하기 어려운 이유로 인식률이 기존의 연구에서 측정한 인식률보다 더 하락했음을 확인하였다[10]. 음성 서비스 경험자들은 자동적으로 음성 앵커를 추출하는 어플리케이션에 흥미를 보이며 보다 적극적으로 실험에 참여한 결과 더 나은 인식률을 보였다. 음성 서비스 무경험자들의 인식률이 음성 서비스 경험자들의 인식률보다 평균적으로 낮은 이유는 경험자들의 음성 입력에 비해 무경험자들의 음성 입력이 익숙하지 못하기 때문으로 예상된다는 기존의 연구 결과가 있다. 음성 어플리케이션을 쉽게 접할 수 있는 충분한 인프라가 구축되었음을 가정했을 때에는 경험자 및 무경험자를 막론한 인식률의 상승으로 서비스를 이용하는 데에 불편함이 없을 것이다. 응답 속도의 측정 결과를 살펴보면, 음성 인식 서비스 경험자와 무경험자의 평균 응답 속도는 최대 1.16초를 초과하지 않았다. 실험 결과로 미루어볼 때 제안하는 시스템이 응답속도가 최대 2초 이내의 기존의 VoiceXML 어플리케이션보다 많은 문자열 데이터의 처리 및 잦은 데이터베이스의 입출력이 이루어짐에도 응답 속도가 더 늦지 않음을 확인하였다[11].

5.2.3 적중률

제안하는 시스템은 사용자에 의해 요구된 정보에서 음성 앵커를 자동으로 추출하여 인식 후보에 등록하는 과정을 수행한다. 시스템이 자동으로 검출 및 등록한 음성 앵커를 사용자가 얼마나 많이 접근했는지, 정확히 접근했는지에 대한 판단의 척도로 적중률을 측정하였다. 제안하는 시스템에서 적중 성공과 적중 실패를 결정하는 단계는 전체 시스템 실행 순서에서 (그림 7)에 도시한 부분으로, 음성 인식률과 상관없이 사용자가 의도하여 발성한 단어가 인식 후보에 존재하는지에 관한 것이다.

(그림 7)에 도시한 적중 성공 및 실패 결정 과정에서 사용자가 의도하여 발성한 단어가 인식 후보에 존재하지 않을 경우, 즉 적중이 실패할 경우에는 사용자가 의도하지 않은 단어로 인식될 수 있으므로 오인식률이 증가한다. <표 3>은 제안하는 시스템의 적중률을 측정하고 평균을 산출한 결과이다. 피험자 A1을 예로 들면 20회의 발성에서 17번 적중하였고 3회 실패하였다. 적중에 실패하였다는 의미는 사용자가 원하는 정보를 정확히 얻지 못했음을 의미한다.

제안하는 시스템의 적중률을 측정한 결과 평균 84.5%의 적중률을 보였다. 적중률 상승을 위한 관건은 서비스 용도에 맞는 정규표현식의 효율적인 작성과 충분한 데이터베이스의 구축에 기반을 둔 음성 앵커의 적절한 검출이다. 적중이 성공했을 시에는 적중한 키워드의 접근 횟수를 누적 기록하여 향후 서비스 정책방향을 결정하거나 서비스를 이용자들의 관심사를 조사하는 것 등에 활용할 수 있다.




(그림 7) 적중 성공 및 실패 결정 과정

〈표 3〉 적중률 측정 결과

| 피험자 | 적중횟수 | 적중률(%) |
|---|---|---|
| A1 | 17 / 20 | 85 |
| A2 | 16 / 20 | 80 |
| A3 | 19 / 20 | 95 |
| A4 | 18 / 20 | 90 |
| A5 | 16 / 20 | 80 |
| B1 | 17 / 20 | 85 |
| B2 | 15 / 20 | 75 |
| B3 | 16 / 20 | 80 |
| B4 | 17 / 20 | 85 |
| B5 | 18 / 20 | 90 |
| 비고 | 169 / 200 | 84.5 |

## 6. 결 론

본 논문에서는 기존의 VoiceXML 다이얼로그 구조인 기계 주도 폼 다이얼로그 구조와 상호 주도 폼 다이얼로그 구조로 설계된 어플리케이션에서 사용자가 자유롭게 웹 자원을 탐색할 수 없음을 인지하고 문제점의 해결을 위해 새로운 다이얼로그 구조인 사용자 주도 폼 다이얼로그 어플리케이션 구조를 제안하였다. 제안하는 시스템의 예로 IT 용어 사전을 내장한 뉴스 서비스를 구현하였고, 음성 인식률, 응답 속도 및 적중률을 측정한 결과 서비스 제공 및 이용에 유효함을 확인하였다. 기존의 다이얼로그 시스템으로 개발된 어플리케이션을 이용하는 것은 사용자가 개발자의 의도에서 벗어날 수 없지만 제안된 다이얼로그 시스템은 사용자의 의도대로 시나리오를 이끌어간다. 이는 VoiceXML의 목적인 음성 인식 및 합성 기술로 웹 자원을 탐색함에 있어서 더 자유로운 인터페이스를 제공할 수 있음을 의미한다.

비주얼 환경의 웹 자원을 구성하는 개념인 하이퍼텍스트 및 하이퍼미디어의 앵커와 노드 구조에 착안하여 설계된 본 시스템은 기존에는 제공할 수 없었던 음성 서비스의 제공을 가능하게 한다. 또한, 자원 탐색 방법이 비주얼 환경의 웹 자원 탐색과 유사하다. 획일적인 음성 서비스에서 벗어나 새로운 유형의 서비스를 제공할 수 있는 방법을 제시한 제안된 시스템이 음성 인식 및 합성 기술을 이용한 서비스를 구상하는 사람들에게 도움이 되길 기대한다.

## 참 고 문 헌


[1] Scott McGlashan, Daniel C. Burnett, Jerry Carter, Peter Danielsen, Jim Ferrans, Andrew Hunt, Bruce Lucas, Brad Porter, Ken Rehor, Steph Tryphonas, "Voice Extensible Markup Language Version 2.0 Specification", http://www.w3c.org/TR/voicexml20, 2004.

[2] Eve Astrid Andersson, Stephen Breitenbach, Tyler Burd, Nirmal Chidambaram, Paul Houle, Daniel Newsome, Xiaofei Tang, Xiaolan Zhu, "Early Adopter VoiceXML", Wrox, 2002.

[3] 박섭형, "음성 웹 어플리케이션 구축을 위한 VoiceXML", 한빛미디어, 2001.

[4] Rahul Ram Vankayala, Hao Shi, "Dynamic Voice User Interface Using VoiceXML and Active Server Pages", LNCS 3841, pp.1181-1184, 2006.

[5] Hemambaradara Reddy, Narayan Annamalai, and Gopal Gupta, "Listener-Controlled Dynamic Navigation of VoiceXML Documents", LNCS 3118, pp.347-354, 2004.

[6] Caccia, G., Lancini, R., Peschiera, G.,"Multimodal browsing using XML/XSL architecture", ITRE2003. IEEE Proceedings of the International Conference on, 2003.

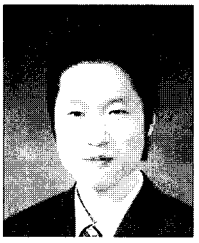
[7] Jeffrey E. F. Friedl, "Mastering Regular Expressions, Third Edition", O'Reilly, 2006.

[8] Eric Gamma, Richard Helm, Ralpg Jhonson, John Vissides, "Design Patterns", Addison-Wesly Publishing Co., 1995.

[9] 국립 국어원, "현대 국어 사용 빈도 조사", 2003.

[10] 권형준, 김정현, 이현구, 홍광석, "콘텐츠 배급을 위한 RSS 기반의 VoiceXML 다이얼로그 시스템", 정보처리학회 논문지 제14-B권 제1호, pp.51-58, 2007.

[11] Min-Jen Tsai, "The VoiceXML Dialog System for the E-Commerce Ordering Service", IEEE Proceedings of the Ninth International Conference, pp.95-100, 2005.


**권 형 준**

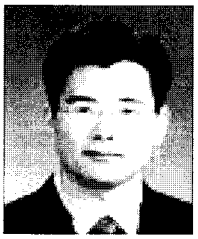
e-mail : katsyuki@skku.edu

2005년 서울보건대학 전산정보처리과 이학사

2005년 (주)블루엠

2006년~현 재 성균관대학교 대학원 정보통신공학부 석사과정

관심분야 : 시스템 통합, 웹 서비스, HCI


**노 용 완**

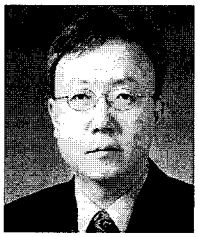
e-mail : elec1004@skku.edu

2001년 남서울대학교 정보통신공학과 학사

2003년 성균관대학교 대학원 정보통신공학부 공학석사

2003년~현 재 성균관대학교 대학원 정보통신공학부 박사과정

관심분야 : 음성인식 및 합성, 신호처리


**이 현 구**

e-mail : lhg@seoil.ac.kr

1988년 성균관대학교 전자공학과 공학사

1991년 성균관대학교 대학원 전자공학과 공학석사

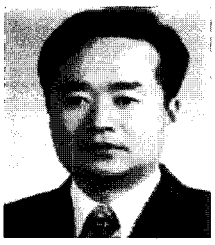
1999년 성균관대학교 대학원 전자공학과 공학박사

1991년~1996년 대영전자기술 연구소

1996년~1998년 부일이동통신 중앙연구소

1988년~현 재 서일대학 정보기술계열 정보통신전공 교수

관심분야 : 통신 및 신호처리, 멀티미디어 통신


**홍 광 석**

e-mail : kshong@yurim.skku.ac.kr

1985년 성균관대학교 전자공학과 공학사

1988년 성균관대학교 대학원 전자공학과 공학석사

1992년 성균관대학교 대학원 전자공학과 공학박사

1990년~1993년 서울보건대학 전산정보처리과 전임강사

1993년~1995년 제주대학교 정보공학과 전임강사

1996년~현 재 성균관대학교 정보통신공학부 교수

관심분야 : 오감 인식, 융합, 재현 및 HCI