

# 음성 사용자 인터페이스 설계 방법론

성신여자대학교 권지혜 · 홍기형

## 1. 서론

공상과학 영화에서나 보아왔던 음성 기술이 유비쿼터스(Ubiquitous) 기술의 비약적인 발전으로 이제 텔레매틱스(Telematics), 게임, 로봇공학, 생활가전 등의 영역에서 우리가 직접 체험할 수 있는 시대가 도래 하였다. 음성 기술은 기계와 인간사이의 상호소통을 인간의 가장 기본적인 자연스러운 의사소통의 수단인 음성을 이용한다는 점에서 사용자에게 큰 매력을 준다. 음성 기술의 핵심은 음성 사용자 인터페이스의 설계이다. 일반적으로 시스템의 사용자 인터페이스는 그래픽 사용자 인터페이스(GUI, Graphical User Interface)에 많은 초점이 맞추어져 왔고, 다양한 응용에서의 다양한 GUI의 설계 방법론에 대해서도 많은 연구가 있어 왔다[1]. 그에 반하여 음성사용자 인터페이스(VUI, Voice User Interface)에 관한 연구는 매우 드물게 진행되었으며, 최근의 음성 기술의 발전과 음성이 전화망 기반 시스템의 인터페이스에서 다양한 휴대기기 및 무선인터넷 등 그 응용 범위가 확대됨에 따라 음성 인터페이스 설계에 관한 연구가 시작되고 있다[2-4].

본고에서는 음성 사용자 인터페이스의 설계가 어떠한 단계를 통하여 비로소 하나의 시스템으로 완성되는지에 대하여 최초로 정리한[2]를 바탕으로 음성 사용자 인터페이스 설계의 방법론에 대하여 소개하고자 한다. 먼저 2장에서는 음성 사용자 인터페이스의 정의 및 특징에 대하여 간략히 기술하고, 3장에서는 음성 사용자 인터페이스 설계 방법론을 구성하는 각 단계에 대하여 소개한다. 4장은 설계의 첫 단계인 VUI 설계를 위한 요구 사항 분석 단계를 설명하고, 5장은 요구 분석 단계의 다음 단계인 실제 설계 단계를 High-level 설계와 상세 설계로 나누어 설명하였다. 6장에서는 설계 결과를 VUI로 개발하는 시스템 개발 단계 및 테스트와 튜닝 단계를 기술하고, 마지막으로 7장에서 결론을 맺는다.

## 2. 음성 사용자 인터페이스의 특징

음성 사용자 인터페이스(VUI)란 사람과 음성언어 어플리케이션이 서로 상호 작용하는 것이다. 음성 사용자 인터페이스는 프롬프트(Prompt), 문법(Grammar), 그리고 대화 로직(Dialog logic)으로 구성되어 있다. 프롬프트는 시스템이 사용자에게 들려주는 합성된 음성 또는 미리 녹음된 음성을 말한다. 문법은 각각의 프롬프트에 대한 사용자의 모든 가능한 응답을 정의하고 있고, 대화 로직은 시스템에 의해 발생할 수 있는 모든 가능한 행위를 정의하고 있다. 이러한 구성요소를 가지고 있는 음성 사용자 인터페이스는 아래의 두 가지 특징을 가지고 있다.

첫째, 음성 사용자 인터페이스의 입력과 출력이 청각을 이용한다는 것이다. 즉, 시스템으로부터의 음성 출력과 사용자로부터의 음성 입력을 소리를 통하여 서로 소통한다는 것이다. 청각을 이용한 음성 인터페이스는 비주얼 웹(Visual-Web) 인터페이스와 달리 지속성 없이 순간적으로 스쳐 지나가는 메시지를 이용하여 사용자와 의사소통을 하기 때문에 설계의 어려움이 많다.

둘째, 음성 사용자 인터페이스는 음성언어에 기반을 두고 있다는 점이다. 음성 사용자 인터페이스와 사용자 간의 의사소통을 이해하기 위해서는 인간 대 인간의 대화방식을 이해하여 그것을 인간과 기계사이의 의사소통에 적용하여야 한다. 인간은 많은 대화의 관습과 인습을 가지고 있고, 또한 대화를 통해 상대방이 다음에 무슨 말을 할 것인지에 대한 기대감을 가진다. 인간의 대화방식에서는 이러한 요소들이 앞으로 전개될 대화의 방향을 결정하는데, 이것을 얼마나 이해하느냐가 성공적인 음성언어를 이용한 인터페이스의 설계에 있어서 가장 중요하다고 할 수 있겠다.

지금까지 언급한 음성 사용자 인터페이스의 특징을 기반으로 하여 이제 음성 사용자 인터페이스 설계의 방법적 단계에 대해서 살펴보도록 하겠다.

### 3. 음성 사용자 인터페이스 설계 방법론

음성 사용자 인터페이스 설계의 단계는 요구 사항 분석, High-Level설계, 상세 설계, 시스템 개발, 테스트, 조율 및 조정의 단계로 크게 여섯 단계로 나눌 수 있다(그림 1).

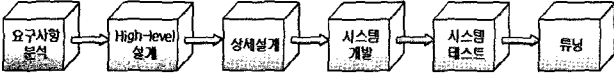


그림 1 음성 사용자 인터페이스 설계 단계

- **요구 사항 분석**: 요구 사항 분석 단계는 시스템 설계에 반영하기 위하여 사업상의 목적, 사용자의 요구, 설계하고자 하는 어플리케이션을 이해하고 구체적으로 사용자 및 비즈니스 요구사항을 정의하는 단계이다.
- **High-level 설계**: High-level 설계 단계에서는 구체적으로 정의된 요구 사항들을 반영하여, 시스템의 구조와 VUI의 설계 원칙 등을 결정하고, 음성 인터페이스의 성격 및 형식을 설계한다.
- **상세 설계**: 상세 설계 단계는 High-level 설계에서 결정한 VUI의 성격 및 형식에 따라, 시스템의 VUI를 구성하는 모든 프롬프트와 콜 흐름을 설계하는 단계이다. 이 단계에서는 모든 가능한 시나리오와 시나리오상의 발생할 수 있는 문제점들에 대해 다룰 수 있도록 대화를 설계해야 한다. 상세 설계 단계에서 구체적으로 다루어야 할 것은 콜 흐름(Call Flow)의 설계, 프롬프트(Prompt) 설계, 사용자 테스트이다.
- **시스템 개발**: 어플리케이션의 실제 생성 단계로서, 설계 단계의 VUI 설계 사항을 코딩, 문법, 오디오 녹음작업을 수행하여 구현한다.
- **시스템 테스트**: 구현한 VUI 시스템에 대하여 원래의 요구사항에 부합하는지를 테스트하는 단계이며, 또한 VUI에 대한 사용성 테스트도 수행한다.
- **튜닝**: 한정된 실제 사용자를 대상으로 시스템을 공개하는 파일럿 테스트를 수행한 후, 시스템에서의 문제점을 보완하는 단계이다. 파일럿 테스트는 실제 서비스 데이터(실제 일을 수행하는 사용자로부터의 데이터)를 사용하여 시스템을 평가하기 위한 첫 관문이다. 서비스 데이터는 주로 인식성능과 문법 범위를 튜닝 하기 위해 사용된다.

## 4. 요구 사항 분석

### 4.1 사업상의 목적 및 요구 사항 수집

먼저 사업상의 목적을 이해하기 위해서는 그 시스템

을 개발하기까지의 주된 사업적 동기가 무엇인지를 파악해야 한다. 가령, 기업의 비용절감을 위함인지, 새로운 서비스의 제공을 위함인지를 파악해야 할 것이다. 또한 새로운 시스템의 도입으로 인하여 다른 시스템과의 조화를 어떻게 이룰 것인지에 대한 고민도 해봐야 한다. 그리고 기업이 추구하는 이미지와 브랜드가 무엇이며 고객에게 무엇을 기여하고자 하는지 고려하여야 한다. 또한 사용자에게 같은 결과물을 줄 수 있는 다른 방법들 또한 모색해봐야 한다. 사업상의 목적을 이해하기 위한 방법으로는 다른 회사 시스템과 사용자간의 접촉수단을 연구하거나, 회사 직원과의 만남을 통하거나, 혹은 경쟁사의 유사 시스템을 평가해보고 그것의 개선점을 자사 시스템에 반영해 보는 것이다.

### 4.2 사용자의 요구 사항 수집

사용자의 요구 사항 수집을 위해서는 누가 시스템을 사용할 것인지, 시스템을 어떻게 사용할 것인지에 대해 알아야 한다. 전자는 시스템을 사용할 대상 사용자가 속해있는 계층, 지적 수준, 유사한 시스템의 사용 경험, 사용자가 바라보는 브랜드이미지 등과 같이 사용자의 특징과 요구를 파악하는 것이다. 후자는 사용자가 시스템을 사용하는 목적이 무엇인지, 시스템의 대상, 용도 등에 대한 정보를 포함한다. 예를 들자면, 기존의 다른 VUI 시스템 사용 경험이 있는지, 없는 지, 또는 모바일(Mobile) 사용자를 위한 시스템이라면 여러 잡음환경에서도 사용할 것인지 등과 같은 정보를 들 수 있겠다.

사용자의 이해를 위해서는 회사 실무자와의 만남을 통하여 서비스 대상에 대한 사전 자료를 제공받을 수 있다. 또한 연구관찰을 통하여 사용자들에 대한 자료를 수집할 수 있다. 그리고 서비스를 이용하는 사용자 대표와의 면담을 통해서 사용자들이 겪는 혼란과 문제점에 대해 알 수 있고 나아가 사용자의 요구를 시스템 설계에 반영 할 수 있다. 그 외에도 사용자를 몇 개의 포커스 그룹으로 나누어 각자의 방식을 토론하도록 한 다든지, 사용자 개개인과의 인터뷰를 한다든지, 우편이나 이메일을 통한 설문조사를 통하는 방법을 이용할 수 있다.

### 4.3 요구 사항 분석 : 어플리케이션(Application)의 기능 도출

사용자의 요구와 사업상의 요구를 수렴하고 분석한 후 그 데이터를 근거로 하여 어플리케이션의 기능들을 명세해야 한다. 이것은 어플리케이션의 각 기능마다 사용자 입력은 어떻게 받고, 사용자의 입력에 대한 결과 값으로서 시스템은 어떤 정보를 반환 할 것이며, 시스

템 내부에서는 어떤 정보를 필요로 하는지를 자세히 정의하는 것이다.

## 5. 설계단계

요구사항의 분석 결과를 바탕으로 VUI의 설계는 그림 2와 같이 High-level 설계와 상세 설계로 나누어 이룬다.

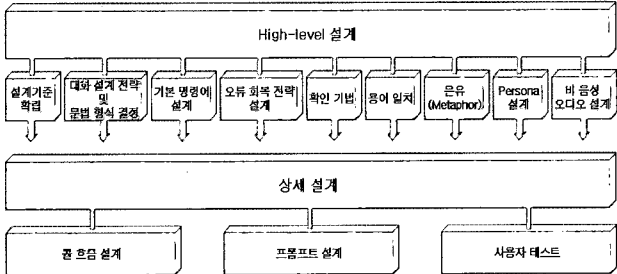


그림 2 설계 단계

### 5.1 High-level 설계 단계

#### 5.1.1 설계기준 확립

High-level 설계 단계 단계에서 가장 먼저 해야 할 일은 요구 사항 분석에서의 사용자와 사업적 목적의 이해를 바탕으로 VUI 설계기준을 정하는 것이다. 예를 들자면, 사용자 측면에서는 편리성을, 사업적인 측면에서는 효율성을, 어플리케이션 측면에서는 정확성을 가장 추구한다고 하자. 그렇다면 이 시스템에서는 편리성, 효율성, 정확성 3개의 설계기준을 설정한 것이다. 설계기준은 하나에서 여러 개까지 정할 수 있으나, 가장 통일성 있는 시스템을 구축하기 위해서는 가급적 설계기준이 하나인 것이 가장 좋다. 이렇게 정리한 설계 기준에 따라 설계의 방향이 결정되는 것이다.

#### 5.1.2 대화 설계 전략(Dialog Strategy) 및 문법 형식의 결정

대화 설계 전략에는 시스템 주도(Directed) 대화 방식과 상호 주도(Mixed-initiative) 대화 방식이 있다. 그림 3은 철도 예약 안내 시스템을 두 가지 방식으로 설계한 것이다.

시스템 주도 대화 방식을 이용하여 여행시스템을 설계한다면, 그림 3(a)와 같이 시스템(현재, 철도청 ARS 예약 시스템[5])은 승차월일, 출발역, 도착역, 승차시각을 차례로 요청하고, 사용자는 시스템이 제시하는 프롬프트에 따라 순서대로 응답하도록 설계해야 한다. 이와 달리 그림 3(b)과 같이 상호 주도 대화 방식으로 한다면, 시스템의 여행일정 요청에 대한 응답으로 사용자는 다양한 응답이 가능하다. 출발역과 도착역만을 먼저 답할 수 있고, 출발시간을 먼저 답할 수 있

시스템 : 열차 잔여석 안내입니다. 승차 월일 네 자리를 눌러주세요.  
 사용자 : 1201(DTMF)  
 시스템 : 출발역을 말씀해 주세요.  
 사용자 : 서울  
 시스템 : 도착역을 말씀해 주세요.  
 사용자 : 부산  
 시스템 : 승차 시각 네 자리를 입력해 주세요.  
 사용자 : 1000(DTMF)

(a) 시스템 주도 대화

시스템 : 열차 잔여석 안내입니다. 여행 일정을 말씀해 주십시오. 예를 들어 서울에서 대구까지 오전 10시 출발이라고 말씀해 주시면 됩니다.  
 사용자 : 서울에서 부산까지  
 시스템 : 몇 시에 출발할 예정입니까?

(b) 상호 주도 대화

그림 3 대화 설계 방식

다. 사용자의 응답에 포함되어 있는 정보의 종류에 따라, 시스템은 부족한 정보를 요청하는 추가 프롬프트를 제공하도록 설계해야 한다. 상호 주도 대화 방식은 시스템 주도 대화 방식보다 시스템과 사용자간의 대화행이 능동적이며 더욱 자연스럽게 진행된다. 상호 주도 대화 방식을 이용하기 위해서는 사용자의 입력에 따라 다음 대화 단계를 선택해야 할 뿐만 아니라 인식 문법도 사용자가 발화할 수 있는 모든 가능성을 포함시켜야 한다.

일반적으로 대화 설계는 사용자의 음성 인식을 위한 문법 설계 방법에 영향을 받는다. 일반적으로 사용되고 있는 문법 방법으로는 규칙기반(Rule-based) 인식 문법과 통계적(Statistical) 인식 문법이다. 규칙기반 문법은 규칙에 의하여 규정된 인식 단어나 문장이 해당 대화에서 행할 수 있는 사용자의 발화 범위를 모두 다룰 수 있을 때 사용된다. 통계적 문법은 학습을 통해서 성능이 좋아질 가능성이 높거나, 설계 기준을 '시스템의 효율성의 최대화'로 설정했을 경우에 사용된다. 일반적으로 상호 주도 대화 방식은 통계적 문법을 이용하는데, 이는 사용자로부터의 다양한 입력을 규칙기반 문법이 다루기에는 한계가 있기 때문이다.

#### 5.1.3 기본명령어(Universal Command) 설계

기본명령어는 시스템이 어떤 대화(dialog)를 진행하고 있는지에 관계없이 사용자가 시스템을 사용하는 전 과정에서 사용할 수 있는 음성 명령어이다. 가장 흔히 사용되는 기본명령어는 '도움말'로써 사용자가 어떤 상태에서든 "도움말"이라고 발화하면, 그 어플리케이션은 현재 대화 상태에서의 상세한 정보를 사용자에게 제공하게 된다. 그 외에도 '초기메뉴', '이전', '종료' 등과 같은 기본명령어들이 있을 수 있다.

### 5.1.4 오류 회복 전략 설계

오류 회복 기법은 음성 인식이 부정확할 경우, 시스템이 어떻게 처리하는지를 다룬다. 보통 인식 오류의 종류는 '거절(reject)'과 '입력시간제한(no speech timeout)'이 있다. '거절'인 경우는 사용자 발화가 인식기의 문법 등에 일치하지 못하는 경우이다. 이 때 시스템은 사용자에게 올바른 답을 유도하기 위하여 다음과 같은 방법을 사용할 수 있다.

- **단계적 프롬프트**: 먼저 Yankelovich, Levow, Marx[6]이 제시한 단계적인 오류 회복방법이 있다. 이 방법은 사용자의 입력에 대해 계속적인 '거절'이 반복될 경우 사용된다. 이 방법은 처음에는 짧은 오류 프롬프트로 시작하여 단계적으로 점점 더 상세한 내용으로 사용자에게 프롬프트 하는 방법이다(그림 4(a)).
- **재 프롬프트**: 다른 방법으로는 인식 거절 프롬프트를 좀 더 간결하게 하는 신속한 재 프롬프트(Rapid reprompt)방법이 있다. 이 방법은 시스템이 오류의 내용에 대해 장황하게 설명하지 않고 간략하게 표현함으로써 사용자들로 하여금 장황한 오류 프롬프트에 지치지 않도록 한다(그림 4(b)). 이 방법은 단계적 프롬프트 보다 효율적인 오류 회복 기법으로 사용 될 수 있다.

Initial Prompt : 승차 월일을 말씀해 주십시오.  
 First Error : 죄송합니다. 정확한 월일을 말씀해 주십시오.  
 Second Error : 죄송합니다. 다시 한번 승차 월일을 확인하여 주십시오. 예를 들어 승차월일이 12월 2일인 경우 '12월 2일' 이라고 말씀해 주시면 됩니다.

(a) 단계적 오류 회복 기법

Initial Prompt : 승차 월일을 말씀해 주십시오.  
 First Error : 죄송합니다.  
 Second Error : 죄송합니다. 예를 들어 '12월 2일'과 같이 정확한 날짜를 말씀해 주십시오.

(b) 신속한 재 프롬프트

그림 4 오류 회복 기법

### 5.1.5 확인(Confirmation) 기법

확인(Confirmation) 기법은 시스템의 신뢰도가 낮거나, 사용자의 입력으로부터 인식 오류를 막기 위해 사용 한다. 확인 기법으로는 그림 5(a)와 같이 항목마다 확인하는 기법과, 그림 5(b)와 같이 하나의 그룹을 묶어 확인하는 기법이 있다. 항목별 확인 과정은 대화의 구성이 지루해 질 수 있으므로 그룹으로 묶어 확인하는 기법을 사용하도록 한다. 그러나 하나의 그룹으로 묶어서 확인 하는 기법인 경우 4개 이상의 항목을 그룹화 하는 것은 피하도록 한다.

시스템 : 출발역을 말씀해 주십시오. (Item1 : 출발역)  
 사용자 : 서울  
 시스템 : 출발역은 '서울'입니다. (Item1 확인)  
 아니면 별표를 눌러 주십시오.  
 시스템 : 도착역을 말씀해 주십시오. (Item2 : 도착역)  
 사용자 : 부산  
 시스템 : 도착역은 '부산'입니다. (Item2 확인)  
 아니면 별표를 눌러 주십시오.  
 시스템 : 승차 시각 네 자리를 입력해 주십시오. (Item3 : 승차시각)  
 사용자 : 1000(DTMF)  
 시스템 : 승차시각은 오전 10시입니다. (Item3 확인)  
 아니면 별표를 눌러 주십시오.

(a) 항목별 확인

시스템 : 출발역을 말씀해 주십시오. (Item1 : 출발역)  
 사용자 : 서울  
 시스템 : 도착역을 말씀해 주십시오. (Item2 : 도착역)  
 사용자 : 부산  
 시스템 : 승차 시각 네자리를 입력해 주십시오. (Item3 : 승차시각)  
 사용자 : 1000(DTMF)  
 시스템 : 요청하신 정보를 확인해 드리겠습니다.  
 출발역 서울, 도착역 부산, 승차시각  
 오전10시입니다. (Item1 ,Item2, Item3 확인)

(b) 그룹으로 확인

그림 5 확인기법

### 5.1.6 용어 일치

만약 VUI 어플리케이션이 웹 사이트와 같은 다른 시스템과 통합되어 사용된다면, 양 시스템간의 용어 일치가 중요하다. 몇몇 사용자는 웹사이트에서의 경험을 이용하여 음성 시스템을 사용하는 경우가 있을 것이다. 이들은 웹사이트에서의 용어, 예를 들어 메뉴의 이름과 구조를 생각하며 VUI 시스템을 탐색해 나갈 것이기 때문에 외부 관련된 다른 시스템이 있다면 반드시 용어를 일치 시켜야 한다.

### 5.1.7 은유(Metaphor)

은유기법은 사용자를 위한 멘탈 모델(Mental Model) -사용자가 시스템과 어떻게 의사소통 하는지를 나타내는 내부적인 모델-의 생성을 돕기 위해 사용자 인터페이스에서 많이 사용되고 있다. 다시 말하면, 사용자가 실생활에서 생각하는 물건이나 행동들을 시스템에 적용하여 형상화 하는 것이다. 예를 들어 증권거래 시스템을 개발한다고 가정하자. 여기서 시스템의 은유를 정의하기 위해서는 먼저 그 시스템의 서비스를 정의해야 한다. 가령 그 시스템이 증권 정보를 고객에게 제공하고, 사용자의 개인 계좌로부터 증권 거래를 수행할 수 있다고 한다면, 사용자와 서비스 사이의 관계를 실제생활에서 거래자와 증권 중개인의 관계처럼 형상화 시켜야 한다. 만약 은유기법이 음성 인터페이스 설계에서 중요하다고 생각된다면, 이 과정은 상세설계 단계에서

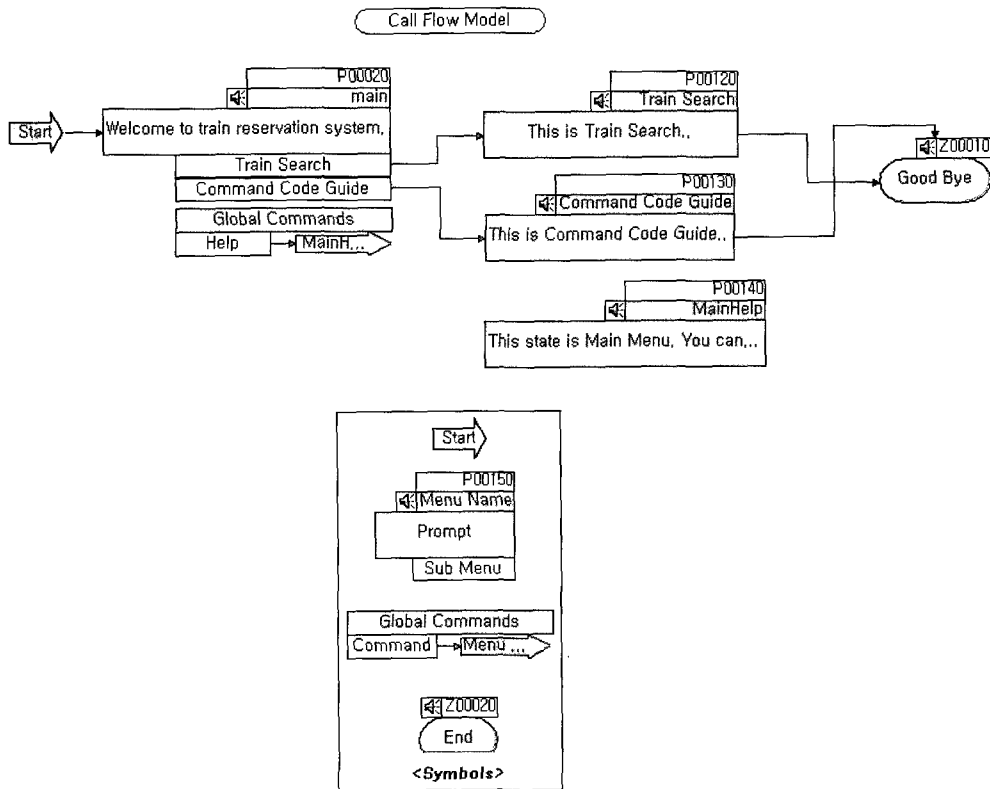


그림 6 IBM WebSphere Voice Toolkit V5.1.0을 이용한 콜 흐름도

행해져야 한다. 왜냐하면, 은유기법은 설계의 다양한 측면에 영향을 미치기 때문이다.

### 5.1.8 Persona 설계

다음으로 시스템의 Persona를 설계해야 한다. Persona는 음성을 통하여 조직의 이미지나 서비스를 나타낼 수 있는 매개물이다. Persona를 설계하기 위해서는 다음과 같은 사항에 대한 고려가 필요하다.

- (1) 먼저 사용자에게 대한 시스템의 역할이 무엇인지 고려해야 한다. 개인 비서처럼 많은 요령과 도움을 줄 것인지, 아니면 지식이나 전문적인 여행정보를 제공할 것인지, 아니면 은행원이나, 전화상담원처럼 매우 사무적인 간단한 질문요청이나 지시만을 제공받을 것인지를 결정해야 한다.
- (2) 다음으로 고려해야 할 사항은 회사의 상표나 이미지도이다.
- (3) 또한 사용자들을 고려해야 한다. 사용자들의 생활 방식, 계층, 태도, 사고방식 등이 어떠한지 파악하고 특징들을 취합하여 각각의 취향에 적합한 Persona를 설계해야 한다.
- (4) 또한 시스템 사용의 빈도수를 고려하여 자주 사용하는 사용자라면 보다 친숙한 Persona로 설계해야 할 것이다.
- (5) 마지막으로 고려되어야 할 사항은 어플리케이션

이다. 시스템의 Persona는 어플리케이션의 내용과 일치해야 하며, Persona의 선택이 일 수행 능력에 있어서 얼마나 효율적인지도 고려해야 한다.

### 5.1.9 비 음성(non-verbal) 오디오의 설계

비 음성 오디오는 시스템에 설계할 음성(프롬프트) 외에 모든 소리를 포함한다. 비 음성 오디오는 적절한 소리음, 가령 배경음이나 효과음의 사용으로 어플리케이션의 사용자 환경을 보다 강화시켜준다. 그러나 너무 많은 소리음의 사용은 오히려 시스템의 효율성을 저하시키므로 주의해야 한다.

## 5.2 상세 설계 단계

### 5.2.1 콜 흐름(Call Flow)의 설계

콜 흐름은 시스템을 구성하는 대화의 기본 구조를 나타내는 것이다. 사용자가 시스템을 어떻게 사용할 것인지, 메뉴는 어떻게 구성할 것인지, 사용자에게 정보를 어떻게 표현할지 효과음은 어디에서 재생할지, 다른 시스템과 어느 부분에서 통합을 할지 등과 같은 사항들을 고려하여 콜 흐름도를 설계해야 한다.

그림 6은 IBM WebSphere Voice Toolkit V5.1.0 [7]을 이용하여 열차 예약 서비스를 나타낸 콜 흐름도이다. 콜 흐름을 설계 할 때에는 어플리케이션의 논리

에 맞도록 자연스럽게 설계하고, 각각의 서브메뉴는 하나의 기능을 수행하는 단위로서 조직한다. 또한 각 메뉴에 대하여 메뉴의 정보를 담고 있는 이름을 짓되, 쿨 흐름도를 구성하는 기호를 각각의 의미에 맞게 규정하고 사용해야 한다. 필요하다면 메뉴의 기능에 대한 프롬프트를 간략하게 추가하도록 할 수 있다

### 5.2.2 프롬프트의 설계

VUI 프롬프트는 음성 언어를 통하여 사용자와 어플리케이션이 서로 의사소통을 하도록 하는 것이다. 프롬프트 설계에서 가장 중요한 것은 사용자 중심의 프롬프트를 설계 하는 것이다. 프롬프트가 보다 자연스러운 형태가 되기 위해서는 먼저 인간의 자연 발생적인 언어인 담화(Discourse) 형식으로서의 대화를 이해해야 한다. Schiffrin은 담화란 문장이나 절 이상의 언어이며 맥락에 의존한다고 정의한다[8]. 이는 인간의 대화가 하나의 구(句)나 단어로만 의미전달이 되는 것이 아니라 문장 전체의 맥락에 의존되어 있다는 것을 말한다. 효과적인 담화의 전달을 위해서는 담화표지(Discourse Marker)나 정보구조를 이용한다.

**담화표지**란 말 단위를 연결하는 요소이다[9]. 담화표지는 '그런데', '그래서', '아직', '사실은' 등과 같이 대화에서 말의 앞뒤 시간관계나 논리관계를 나타내어 문장의 다음에 무슨 내용이 올지 미리 예측하게 해준다. 그림 7은 담화 표지를 문장에 사용하지 않은 경우와 사용한 경우이다.

- (1) 출발역을 말씀해 주십시오.
- (2) 도착역을 말씀해 주십시오.
- (3) 승차시각 네 자리를 입력해 주십시오.

(a) 담화표지를 사용하지 않은 경우

- (1) 먼저, 출발역을 말씀해 주십시오.
- (2) 자, 다음으로, 도착역을 말씀해 주십시오.
- (3) 마지막으로, 승차월일 네 자리를 입력해 주십시오.

(b)담화 표지를 사용한 경우

그림 7 담화 표지

그림 7(b)와 같이 담화표지를 프롬프트에 사용하게 되면 사용자는 '먼저'라는 담화표지를 사용함으로써 앞으로 질문이 계속 될 것이라는 것과, '마지막으로'라는 담화 표지로 인하여 이 질문이 마지막임을 예측할 수 있다.

**정보구조**란, 대화의 내용을 신(New)정보와 구(Old)정보로 나눌 수 있는데, 이 중 상대방에게 전달하고자 하는 중심 정보 즉, 신정보의 위치에 따라 사용자에게 전달되는 효과가 차이가 난다. 이는 대부분의 사용자는 한꺼번에 다량의 정보를 바로 받아들이는 데에 기억력

의 한계가 있으며 최후에 들은 것을 더 오래 기억하는 성향이 있다. 그렇기 때문에 VUI 설계자는 초점이 되는 신정보의 위치를 문장의 마지막에 위치시키도록 한다.

담화는 사용자의 사회적 문화배경에 따라 언어 양상이 다르게 표현된다. 이를 언어학에서 **사용력(Register)**이라고 한다. 사용력은 수단(Mode), 장(Field), 태도(Tenor)로 분류 한다[10]. 수단(Mode)은 담화의 수단이 '말(음성)'인지 '글'인지를 나타낸다. 장(Field)은 담화의 목적과 내용에 대한 것이다. 마지막으로, 태도(Tenor)는 누가 누구에게 말하는가와 같이 담화 참여자의 관계에 대해 나타낸다. VUI 설계자는 어플리케이션에 적절한 사용력(Register)를 고려하여 설계한 대화 곳곳에 일관되도록 적용시켜야 한다.

때때로, 대상 사용자의 직업에 따라 대화내용에 **전문용어**의 사용이 빈번한 경우가 있다. 만약 시스템 사용자 대상이 특정분야의 전문가 그룹이라면, 프롬프트 설계 시 전문용어의 사용을 고려해 볼 수 있다. 그러나 사용자 대상이 일반인이라면 전문용어의 사용은 가급적 지양하고 모두에게 친숙한 용어를 사용하도록 한다. 아래의 예는 의약품 설명에 관한 전문용어의 사용 예시이다. 일반인을 대상으로 한 시스템에서는 그림 8(b)에서와 같이 순화된 표현을 사용하도록 한다[11].

- 000은 서방형 제제이므로 정제를 으깨거나 씹거나 녹이지 말고 그대로 삼키십시오.

(a) 전문용어의 사용

- 000은 체내에서 서서히 분산되는 성질을 가진 약이므로 알약을 으깨거나 씹거나 녹이지 말고 그대로 삼키십시오.

(b) 전문용어의 순화된 표현

그림 8 전문용어

프롬프트 설계 시 마지막으로 고려해야 할 것은 대화의 **협력원칙(The Cooperative Principle)**이다. 협력원칙은 대화가 진행되는 각 단계에서 대화의 방향이나 목적에 의해 요구되는 만큼 기여를 해야 한다는 원칙이다[12]. 그림 9는 협력원칙에 위배되는 예이다. 시스템의 첫 프롬프트는 포인트 적립이라는 서비스를 웹사이트에서만 서비스하는 것처럼 사용자가 착각할 수 있는 여지를 준다.

시스템 : 고객님의 안녕하십니까. 포인트 적립을 지금 웹사이트에서 하실 수 있습니다.  
 시스템 : 원하시는 서비스를 말씀해 주십시오. "포인트 적립", "포인트 충전"...

그림 9 협력원칙에 위배되는 경우

협력원칙은 양(Quantity), 질(Quality), 관계(Re-lation), 방법(Manner)의 원칙으로 볼 수 있다. 양은 필요한 양의 정보만을 전달하라는 의미를 가지고 있고, 질은 잘못된 정보는 제공하지 말라는 것이며, 관계는 연관된 내용이어야 한다는 것이고, 마지막으로 방법은 불분명한 표현은 하지 말며, 모호성을 피하고, 명료해야 하며, 순서 있게 표현하라는 것이다. 결국 협력원칙은 사용자에게 적절한 양의 정보를 모호하지 않고 분명하게 전달하라는 것을 말하고 있다.

### 5.2.3 사용자 테스트

여기에서의 사용자 테스트는 앞서 설계한 콜 흐름 설계와 프롬프트 설계를 위한 테스트이다. 시스템 설계 초기에 사용자테스트를 수행하면 시스템에 발생할 수 있는 위험요소를 사전에 예방할 수 있다.

## 6. 시스템 개발, 테스트, 튜닝 단계

그림 10은 시스템 개발, 테스트, 튜닝의 세부 단계를 보이고 있다.

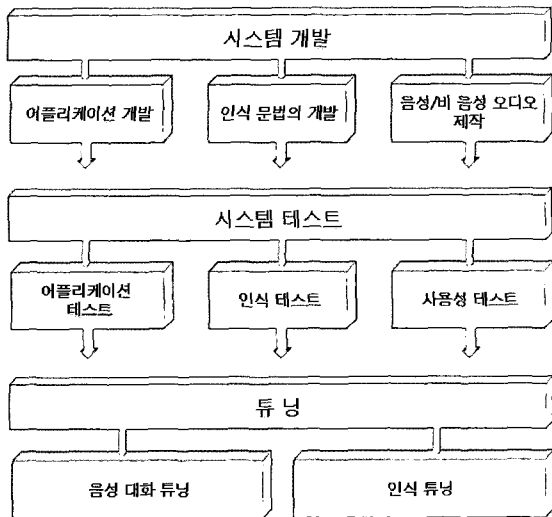


그림 10 시스템 개발, 테스트, 튜닝 단계

### 6.1 시스템 개발

#### 6.1.1 어플리케이션 개발

VUI 어플리케이션의 개발은 설계한 대화를 코딩하고 다른 소프트웨어 시스템과 데이터베이스와 통합을 하는 것이다. 어플리케이션의 개발은 여러 가지 프로그래밍 언어로 구현될 수 있는데, 음성기반의 어플리케이션 개발은 음성 플랫폼을 이용한다는 특성에 맞게 VoiceXML[4, 13]을 이용하여 음성 대화 코딩을 구현할 수 있다.

#### 6.1.2 인식 문법의 개발

인식 문법의 개발은 실제 사용자가 말할 수 있는 가

능한 모든 범위를 다룰 수 있도록 해야 한다. 인식 문법 개발의 과정은 규칙기반 문법 방법과 통계적 문법 방법이 서로 다르다.

규칙기반 문법방식은 완전한 규칙에 정의된 대로 문법을 생성하는 것이다. 설계자는 문법을 생성할 때 문법에 중심 정보(Slot)와 주변정보(Filler)를 분리하여 작성하여야 한다. 주변정보는 중심정보를 감싸고 있는 단어나 구(句)이다. 문법을 생성하는 규칙은 ABNF, XML Form형식으로 나타낼 수 있으며, 표 1과 그림 11에서처럼 플랫폼마다 제공되는 문법 생성 규칙을 이용할 수도 있다. 표 1은 Nuance의 문법 명세 언어(Grammar Specification Language)에 사용되는 연산자를 나타내고 있다[14]. 문법 명세 언어는 '문법 이름'과 '문법 설명'의 규칙을 가지고 있다. 그림 11은 .GETSTATION 문법에 3개의 하위 문법(PREFILLER, CITY, POSTFILLER)을 정의하는 문법을 나타내고 있다. 이 문법에 유효한 문장으로는 "아 부산 부탁합니다" 혹은 "저기 서울이요"이다.

표 1 GSL을 위한 문법 연산자

| Operator | Expression  | Meaning                                      |
|----------|-------------|--|
| ()       | (A B C...D) | A followed by B followed by C followed by..D |
| []       | [A B C...D] | One of A or B or C or ..D                    |
| ?        | ?A          | A is optional.                               |
| +        | +A          | A occur one or more times.                   |
| *        | *A          | A occur zero or more times.                  |

```
.GETSTATION(?PREFILLER CITY ?POSTFILLER)
PREFILLER[아
    ?저기
]
CITY [[(부산)
      (서울)
]
POSTFILLER[부탁합니다
            이요
]
]
```

그림 11 GSL을 사용한 문법의 예

통계적 언어 모델은 사용자의 입력이 일정한 규칙기반 문법으로 다루기 어려울 정도로 광범위 할 때 사용된다. 통계적 언어 모델을 위한 문법의 개발은 사용자가 입력할 수 있는 데이터를 수집하여 반복적인 학습을 통하여 접근한다. 설계자는 사용자들의 언어를 훈련과정을 거쳐서 통계적 언어모델을 생성한다. 이 때 사용되는 방법이 Markov의 순서모델(order model)이며 다른 이름으로 N-gram[15]이라고도 한다. N-gram 모델은 어플리케이션에서 대용량 어휘를 지원하기 위해 사용된다. N-gram은 공백을 제거한 주어진 문자열로부터 N개 문자의 연쇄를 확률적으로 표현

한 것이다. N=1일 때 unigram, N=2일 때 bigram, N=3일 때 trigram이라고 하는데 이론적으로 N의 차수가 높을 때 어떤 단어가 다음에 위치할지에 대한 학습 성능이 우수해진다. 그러나 높은 차수의 모델은 고려해야 할 파라미터가 방대해 지기 때문에 대개 trigram이 실제 사용되고 있다.

### · 6.1.3 음성/ 비음성 오디오 제작

오디오의 제작은 시스템에 사용할 모든 프롬프트 및 비 음성 오디오를 녹음·제작하는 작업이다. 프롬프트를 녹음하기 위해서는 시스템의 목적과 사용자의 요구 사항에 부합하는 음성 행위자(Voice Actor)를 선택해야 한다. 음성 행위자는 시스템과 사용자 사이에서 실제로 사용하는 프롬프트를 낭독해야 하기 때문에 적절한 Persona와 억양, 말씨를 사용하여야 한다. 이를 위해서는 음성 행위자가 낭독할 스크립트의 내용을 충실히 작성하여야 한다. 스크립트의 내용에는 아이템 번호, 프롬프트 할 내용, 각 프롬프트에 대한 보충설명, 녹음될 파일이름 등을 기입한다. 보충설명에는 프롬프트 할 내용의 발음을 표시하거나 억양이나 강세를 표시하도록 하여 음성 행위자가 낭독하기 편하도록 한다. 스크립트는 음성 행위자의 한눈에 스크립트의 내용 전체가 보이도록 하며, 큰 글자를 이용하도록 한다. 또한 내용의 간격이 조밀하지 않도록 해야 하고, 긴 문장은 여러 개의 문장으로 분리하여 음성 행위자의 호흡에 맞도록 한다.

## 6.2 시스템 테스트

### 6.2.1 어플리케이션 테스트

어플리케이션 테스트는 음성 대화의 설계가 계획대로 짜임새 있게 구현 되었는지, 사용자로부터의 콜을 시스템이 적절히 수용하는지를 테스트한다. 어플리케이션 테스트는 다시 음성 대화 테스트, 시스템 QA 테스트, 작업부하 테스트로 세분화 된다.

**음성 대화 테스트**는 시스템이 설계한 대로 음성 대화를 정확히 구현되었는지 확인하기 위해 수행된다. 음성 대화 테스트는 모든 대화의 상태를 테스트하며 사용자의 입력에 올바른 프롬프트가 재생되는지, 사용자의 입력에 따라 시스템이 올바른 방향으로 수행되는지를 살펴본다. 특히 기본명령어는 제대로 수행 되는지, 오류 처리는 제대로 수행되는지, 등을 테스트 한다.

**시스템 QA 테스트**는 다른 시스템과 통합하여 테스트를 수행하는 것으로 모든 통합 환경에서 모든 조건을 테스트 한다.

**부하테스트**는 사용자가 가장 많이 사용하는 시간대에서 부하가 최고점에 이를 때 시스템이 이를 얼마나

효율적으로 대처할 수 있는지를 테스트 한다. 부하테스트는 일반적으로 콜 센터나 서비스 업자에게 소프트웨어를 설치한 후에 시행하며, 동시에 많은 사용자가 접속하여 테스트한다.

### 6.2.2 인식 및 사용성 테스트

인식 테스트는 허용 범위 안에서 음성 인식이 얼마나 잘 되는지를 테스트 한다. 사용성 테스트는 시스템을 공개하기 전에 수행하며 시스템 운용 중 발생하는 문제점(예: 사용자로 인한 시스템 지연문제, 잡음 환경에서의 반복 오 인식 등)을 발견하기 위해 시행한다.

## 6.3 튜닝(Tuning)

테스트 단계의 결과를 이용하여 보다 나은 성능을 확보하기 위하여, 다음과 같은 시스템 튜닝 작업을 수행한다.

### 6.3.1 음성 대화 튜닝

음성 대화 튜닝은 사용자와 시스템 간에 커뮤니케이션이 보다 효율적으로 진행될 수 있도록 조정하는 것이다. 음성 대화 튜닝은 콜 모니터링, 콜 로그 분석을 위한 유틸리티를 사용하거나, 사용자 경험 연구로 음성 대화 성능을 평가할 수 있다.

콜 모니터링은 시스템 성능을 이해하기 위해 콜을 모니터 하는 것이다. 이것은 100개의 콜을 무작위로 선택하여 실제 콜을 듣거나 녹음하여 시스템이 어떻게 진행하는지를 보는 것이다. 콜 모니터링은 시스템 측과 사용자 측 대화를 모두 들을 수 있고 Barge-in과 같은 문제를 좀 더 쉽게 진단할 수 있다는 장점이 있다.

콜 로그는 시스템을 사용한 사용자의 음성 기록을 지칭하며, 이 기록을 분석하여 음성 대화에서 비효율적인 부분이 어디인지 식별하는데 유용하게 사용된다.

사용자 경험 연구는 파일럿 기간 동안 시스템을 사용한 사람들에게 설문지를 보내어 조사한다.

### 6.3.2 인식 튜닝

인식튜닝은 인식 문법에 대한 인식의 정확성을 높이기 위한 것이다.

인식튜닝의 첫 단계는 인식 문법 성능을 측정하는 것이다. 이것은 사용자의 발화에 대해 인식기의 인식 결과를 서로 비교해 봄으로써 이루어진다. 그 결과는 In-grammar와 Out-of-grammar로 나눌 수 있다. In-grammar는 문법에 정의된 것을 사용자가 말하는 경우이고, Out-of-grammar는 문법에 정의되지 않은 것을 사용자가 말하는 경우이다. In-grammar는 경우에 따라 세 가지로 분류된다. 인식기가 정확한 답을 반환하는 Correct accept, 인식기가 틀린 답을 반환하는 False accept, 인식기가 문법의 어떤 경로에서도



입력 값에 일치되는 답이 없을 때 답을 반환하지 않고 거절하는, False reject로 분류된다. Out-of-grammar는 두 가지로 분류된다. 인식기가 사용자의 입력에 정확히 거절되는 Correct reject와 문법의 정의에 없는 입력에 대한 잘못된 답을 반환하는 False accept로 분류된다. 인식 성능은 이 값들의 비율에 따라 결정되며, 인식기의 인식 파라미터를 조정함으로써 인식 성능이 개선될 수 있다. 조정할 수 있는 인식 파라미터의 예로는, 인식 거절(rejection) 임계값이 있다. 인식 거절 임계값은 인식 거절이 되는 인식결과의 신뢰도 값을 의미하며, 이 임계값이 낮으면 인식 거절이 적게 발생한다.

## 7. 결 론

본 고에서는 음성 사용자 인터페이스(VUI) 설계 방법론을 소개하고, VUI 설계 및 개발의 각 단계에 대하여 알아보았다. 사용자 중심의 VUI 설계를 위해서는 설계자가 사용자의 다양한 경험을 바탕으로 한 데이터의 수집을 기반으로 설계에 임해야 한다. VUI 설계 단계에서 고려해야 하는 중요한 원칙은 다음과 같다[2].

- 첫째, 인식 부하를 최소화한다.
- 둘째, 대화적 관습을 수용한다.
- 셋째, 효율성을 최대화한다.
- 넷째, 명료성을 최대화한다.
- 다섯째, 높은 정확성을 보장하도록 한다.
- 여섯째, 오류로부터의 회복을 적절히 수행한다.

아직까지 한국어에 적합한 음성기반 사용자 인터페이스의 설계에 대한 연구는 그다지 활발하지 않다. 기본적인 VUI 설계 방법론이나 요구사항 분석, high-level 및 상세 설계, 개발, 테스트 및 튜닝의 단계는 이미 정립된 영어권의 VUI 개발 방법론을 사용하면 큰 문제는 없을 것으로 보인다. 그러나 한국어 및 우리 문화가 가진 고유한 특징을 high-level 설계와 상세 설계의 세부 기법에 반영되어야 할 것이다.

음성 인식 및 합성 기술이 지속적으로 발전하고 있으며, 유비쿼터스 환경의 발전에 힘입어 음성 기반 멀티모달 인터페이스 기술의 필요성이 증대되고 있다. 이에 따라, 우리말 음성 사용자 인터페이스 설계에 관한 본격적인 연구가 필요하며, 향후 이 분야의 확대 및 발전이 예상된다.

## 참고문헌

[1] David Redmond-Pyle and Alan Moore,

Graphical User Interface Design and Evaluation Guide, Prentice Hall, 1995.

- [2] Michael H. Cohen, James P. Giangolam, Jennifer Balogh, Voice User Interface Design, Addison-Wesley, 2004.
- [3] Michael F. McTear, "Spoken dialogue technology," ACM Computing Surveys(CSUR), Vol.34, Issue.1, pp.90~169, 2002.
- [4] Chetan Sharma, Jeff Kunins, VoiceXML: Strategies and Techniques for Effective Voice Application Development with VoiceXML 2.0, Willey Computer Publishing, 2002.
- [5] 철도청 ARS 예약시스템, 060-700-1188(전화).
- [6] Nicole Yankelovich, Gina-Anne Levow, Matt Marx, "Designing SpeechActs: Issues in Speech User Interfaces," Conference on Human Factors in Computing Systems, CHI '95 Proceedings, pp369~376, 1995.
- [7] Voice Toolkit V5.1 for WebSphere Studio Web Site, [http://www-306.ibm.com/software/pervasive/voice\\_toolkit/](http://www-306.ibm.com/software/pervasive/voice_toolkit/), 2004.
- [8] Schiffrin, D, Approaches to discourse, Cambridge University Press, 1998.
- [9] 김태엽, 담화 표지 되기와 문법화, 우리말글26집, 우리말글학회, pp61~80, 2002.
- [10] Halliday, M.A.K, An introduction to functional grammar, 2nd edition, Edward Arnold, 1994.
- [11] 남기심, 제품 설명서의 문장 실태 연구 1, 국립국어연구원, 2002.
- [12] Grice, H.P, Logic and conversation, In P.Cole and J.L morgan, eds., Speech acts, pp41~45, Academic Press, 1975.
- [13] Voice Extensible Markup Language(VoiceXML) 2.1 Web Site, <http://www.w3.org/TR/2005/CR-voicexml21-20050613>, 2005.
- [14] Nuance Speech Recognition System Version 8.5: Grammar Developer's Guide, Nuance Communications Inc, 2003.
- [15] Michael K. Brown, Andreas Kellner, Dave Raggett, "Stochastic Language Models (N-Gram) Specification," <http://www.w3.org/TR/2001/WD-ngram-spec-20010103>, 2001.

---

권 지 혜



2003. 2 성신여자대학교 컴퓨터정보학부  
(학사)  
2004. 2~현재 성신여자대학교 교육대학원  
전자계산교육(석사)  
관심분야 : 멀티모달 인터페이스, Voice-  
XML, CCXML  
E-mail : jkwon@sungshin.ac.kr

홍 기 형



1985. 2 서울대학교 컴퓨터공학과(학사)  
1987. 2 한국과학기술원 전산학과(석사)  
1994. 2 한국과학기술원 전산학과(박사)  
1994. 2~1998. 2 한국전자통신연구원  
데이터베이스팀 선임연구원  
1998. 3~현재 성신여자대학교 미디어  
정보학부 부교수  
관심분야 : 음성응용, XML, 멀티모달 인  
터페이스, 멀티미디어 DB  
E-mail : khhong@sungshin.ac.kr

---