

# 안드로이드 기반 보이스 커뮤니케이션 시스템 구현

이지현\*, 박영호\*

\*숙명여자대학교 멀티미디어학과

e-mail: cong891@naver.com, yhpark@sookmyung.ac.kr

## An Implementation of Voice Communication System based-on Android Platform

Ji-Hyeon Lee\*, Young-Ho Park\*

\*Dept of Multimedia Science, Sookmyung University

### 요 약

최근 스마트폰의 사용률이 증가하고 다양한 어플리케이션이 등장하고 있다. 그 중에서도 스마트폰을 활용하여 사용자간 커뮤니케이션을 나눌 수 있는 어플리케이션에 대한 관심이 급증하고 있다. 따라서, 본 논문에서는 스마트폰 상에서 채팅방을 생성하여 관심사가 같은 여러 사용자와 보이스 채팅을 나눌 수 있는 보이스 커뮤니케이션 시스템을 제안한다. 안드로이드 플랫폼 기반의 스마트폰에서 채팅방을 생성하고, 이곳에 제한된 인원이 모여서 대화가 가능하다. 본 시스템을 통하여 사용자들은 마음이 맞는 사람들과 음성으로 간편하게 커뮤니케이션할 수 있을 것으로 기대한다.

### 1. 서론

2009년 12월 대한민국에 스마트폰인 애플의 아이폰이 도입된 이후로 다양한 회사의 스마트폰이 등장하고 있다. 이러한 스마트폰은 휴대가 간편하고 데이터 접근이 용이하여 그 사용량이 빠르게 증가하고 있고 이에 힘입어 다양한 어플리케이션이 개발되고 있다. 최근 등장한 어플리케이션의 순위를 매긴 결과, 유료 어플리케이션으로는 '와츠앱 메신저(WhatsApp Messenger)'가 무료 어플리케이션으로는 카카오톡이 차지하고, 리서치기관인 '가트너'에서 2011년 IT분야 전망치에서 '소셜 커뮤니케이션과 협업'이 중요할 것이라고 판단하면서 사람들이 스마트폰을 활용한 커뮤니케이션에 관심이 높다는 것이 확인되었다.[1][2]

현재 소셜 커뮤니케이션 관련 어플리케이션에서는 텍스트나 음성을 이용하여 사용자간 커뮤니케이션을 나눌 수 있도록 한다. 그 중 음성을 이용한 커뮤니케이션을 제공하는 어플리케이션은 대화 상대자 선정 방식에 따라 크게 채널이나 전화번호부를 기반으로 나뉘어 제공되고 있다.

그러나, 현재 상용화 되어있는 어플리케이션은 문제점을 가지고 있다. 채널 기반 서비스의 경우, 채널에 입장한 서로 전혀 관련이 없는 불특정 다수의 사용자들 간에 대화가 이루어져 대화에 대한 흥미가 떨어지고 많은 수의 사용자들이 모였을 경우 발언권에 제한이 있어 사용자간 소통에 어려움이 있다. 또한, 전화번호부 기반 서비스는 사용자가 전화번호부에 등록해놓은 사용자들과만 대화가 가능하다는 제한이 있고 상대방의 상황을 고려하지 않고 회의나 수업 시간에도 바로 대화가 전달되어 많은 문제를 일으킨다.

따라서 본 논문에서는 이러한 문제점을 보완하여 관심사가 같은 여러 사용자와 음성으로 대화를 나눌 수 있는 보이스 커뮤니케이션 시스템을 제안한다.

본 시스템의 장점은 다음과 같다. 첫째, 기존의 불특정 다수와 대화를 나누었던 것을 보완하여, 채팅방을 생성할 수 있어 관심사가 같은 사람들이 모여 대화를 나눌 수 있도록 하였다. 둘째, 기존 서비스에서 문제가 될 만한 발언권을 반영했다. 한 채팅방에 입장할 수 있는 인원수에 제한을 설정하여 사용자간 소통이 원활하게 하였다.

이런 장점들을 취합하여 안드로이드라는 새로운 스마트폰 플랫폼에서 관심사가 같은 다양한 사람들과 음성을 이용하여 간편하게 커뮤니케이션을 하는 어플리케이션을 개발하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 보이스 기반 커뮤니케이션 시스템을 소개 및 분석한다. 3장에서는 제안하는 시스템을 소개하고 개발환경과 전반적인 시스템의 구현 및 기능별 알고리즘에 대하여 설명한다. 4장에서는 본 시스템의 구현 결과를 보인다. 5장에서는 본 논문의 결론을 맺는다.

### 2. 관련 연구

본 장에서는 기존에 있는 스마트폰 보이스 기반 커뮤니케이션 시스템에 대하여 설명한다. 2.1절에서는 TiKL, 2.2절에서는 iPTT를 소개 및 분석한다.

## 2.1 TiKL (Touch to Talk)

TiKL[6]은 안드로이드 커뮤니케이션 어플리케이션이다. 전화번호부를 기반으로 하여 사용자가 전화번호부에 저장해놓은 사람들과 대화를 나눌 수 있다.

TiKL에서는 다음과 같은 기능을 제공한다. 첫째, 전화번호부 중에서 대화를 나누고 싶은 사람들을 선택한다. 둘째, 버튼 터치 한 번으로 사람들과 간편하게 대화를 나눌 수 있다.

하지만, 전화번호부에 등록된 사람들과만 대화가 가능하도록 제한되어 있다. 또한 특별한 설정을 하지 않으면 상대방이 회의나 식사시간과 같이 대화를 수신하고 싶지 않은 상황에도 이를 고려하지 않고 바로 대화가 전달된다는 문제가 있다[4].

## 2.2 iPTT (Push-to-talk)

iPTT[5][7]은 아이폰 커뮤니케이션 어플리케이션이다. 여러 사람이 실시간으로 함께 대화를 나눌 수 있도록 서비스를 제공하고 있다. 또한, 채널을 기반으로 하여 장소에 구애받지 않고 채널을 선택하여 전세계 사람들과 대화를 나눌 수 있다.

iPTT는 공개모드와 속삭임모드를 제공한다. 공개모드에서는 채널을 선택하여 장소에 구애받지 않고 전세계 사람들과 대화를 나눌 수 있다. 속삭임 모드에서는 같은 채널상의 사용자를 선택하여 일대일 사적인 대화를 나눌 수 있다.

하지만, 채널에 입장한 불특정 다수의 사용자들 간에 대화가 이루어져 발언권에 제한이 있어 사용자간 소통이 제대로 이루어지지 않는다.

## 3. 보이스 커뮤니케이션 시스템의 구현

본 장에서는 본 논문이 제안하는 시스템의 개요와 개발 환경 및 구현에 대해 설명하고 전반적인 기능별 알고리즘을 소개한다.

### 3.1 개요

보이스 커뮤니케이션 시스템은 채팅방을 형성하고 그 안에서 사용자간 보이스 채팅을 나누는 모바일 커뮤니케이션 시스템이다.

사용자는 모바일 클라이언트를 이용하여 회원 가입을 하고 채팅방을 형성, 관심사별로 모여 다른 사용자들과 대화를 나눈다. 서버는 클라이언트로부터 받은 정보를 처리한다.

본 시스템은 클라이언트 간 원활한 소통을 위해 채팅방 기능을 제공한다. 채팅방은 사용자가 생성하는 공간같은 것으로 관심사별로 사용자들을 모일 수 있게 해준다.

보이스 커뮤니케이션 시스템은 음성을 이용하여 간편하게 대화를 나누므로 모바일 환경에서의 간편한 커뮤니케이션의 활용을 높일 수 있다.

### 3.2 개발 환경

보이스 커뮤니케이션 시스템은 크게 서버와 클라이언트 두 파트로 나누어져 개발이 되었다. 클라이언트는 eclipse-helios와 Android 2.2 버전으로 테스트 단말기 갤럭시A의 스크린 사이즈 800\*480에 최적화되도록 개발하였다, 서버의 데이터베이스는 MS-SQL Server 2005를 사용하였고, 윈도우 기반 웹 서버인 IIS를 설치하였다. 그리고 데이터베이스와 연동하기 위해 IIS 환경에 최적화된 스크립트 언어인 ASP를 EditPlus 3.2 버전을 이용하여 구현하였다[3].

### 3.3 서버와 클라이언트 구현

본 절에서는 시스템의 구현을 서버와 데이터베이스, 클라이언트로 나누어 설명한다.

#### 3.3.1 서버

본 절에서는 서버의 설계에서 핵심이 되는 구현을 소개한다.

첫째, 서버는 ASP 페이지를 통하여 클라이언트가 전달한 정보를 데이터베이스에 저장하거나 데이터베이스로부터 얻은 정보를 클라이언트에 전달한다. [그림 1]은 서버에서 클라이언트로 정보를 전달하는 기능이다. ASP 파일을 통해 데이터베이스에 연결하고 클라이언트에서 받은 Request객체의 내용을 이용하여 SQL문을 전송한다. 그리고 Response객체에 XML형태로 데이터를 작성하여 클라이언트에게 전달한다. 둘째, 본 시스템의 소켓 프로그래밍은 String 타입을 기반으로 작동한다. 이 때, 서버는 클라이언트로부터 넘어온 메시지를 소켓에 접속한 다른 클라이언트들에게 넘겨준다.

```

<%
Set ConnDB = Server.CreateObject("ADODB.Connection")
ConnDB.open "DSN=sql_sic;UID=sa;PWD=mmpri;"
%>

<%
Set objRs=Server.CreateObject("ADODB.RecordSet")
objRs.Open "SELECT * FROM record WHERE roomId="
& Request("roomId") & " order by no DESC",ConnDB,1

Response.Write "<?xml version='1.0' encoding='UTF-8'?">%"

</lists>
<% For j = 1 to 2 %>
<data>
<name><%=objRs("name")%></name>
<status><%=objRs("status")%></status>
</data>
<%
objRs.MoveNext

If objRs.EOF Then
Exit For
End If
Next
%>
<%
ConnDB.Close
Set ConnDB = Nothing
Set objRs = Nothing
%>
</lists>

```

[그림 1] 정보를 XML형태로 작성하는 기능

### 3.3.2 데이터베이스

테이블은 회원 정보 관련 테이블과 채팅방 정보 관련 테이블로 나누어지며 총 4개의 테이블이 존재한다. 먼저 회원 정보 관련 테이블로는 가입 시 입력한 회원의 기본 정보를 담고 있는 member 테이블과 로그인한 회원의 위치 및 입장한 채팅방 정보를 담고 있는 loginlist 테이블이 있다. 다음으로 채팅방 정보 관련 테이블로는 채팅방 생성 시 입력한 정보를 담고 있는 community 테이블과 회원 간의 대화 정보를 담고 있는 record 테이블이 있다.

모든 회원 정보 관련 테이블은 member 테이블의 Primary Key인 tel을 Foreign Key로 가지게 된다.

### 3.3.3 클라이언트

본 절에서는 클라이언트의 설계에서 핵심이 되는 네 가지를 소개한다.

첫째, 안드로이드 구성요소 4가지 중에 서비스와 액티비티를 사용한다.[9] 시스템에서는 액티비티를 사용하여 화면을 구성한다. 또 서버와의 소켓 통신을 위하여 본 시스템에서는 서비스를 사용한다.

둘째, 서버에게 메시지를 패킹해서 보낸다. 사용자에게 입력 받은 데이터들을 String으로 변환한 후 모든 String을 하나로 패킹한다.

셋째, 서버로부터 받은 메시지를 파싱하여 처리한다. 클라이언트는 서버로부터 넘어온 메시지를 '/' 단위로 파싱하여 작업을 처리한다[8].

넷째, ASP 파일을 통해 서버와 통신한다. 여기에서는 클라이언트의 cmsHTTP와 Util 클래스가 사용된다. cmsHTTP클래스는 서버와 통신을 관리하며 결과값을 받아온다. 서버로 보낼 내용은 HttpPost에 장착한 후 HttpClient를 통하여 전송한다. 그리고 서버로부터 받은 내용은 HttpResponse에 저장하여 스트링타입으로 변환 후 결과 값으로 리턴한다. Util.java는 데이터 타입 변환 메소드를 모아놓은 클래스이다. 서버와 통신 시에 xml2HashMap() 메소드를 사용하여 XML문자열을 DOM해독 방식으로 해독, 해독 결과 값의 각 노드명과 노드 값을 HashMap형태("노드명[레코드번호]=노드값")로 저장한 후 리턴한다. 서버에 정보를 전달할 시에는 클라이언트에서 입력한 정보를 ArrayList로 정의한 후 cmsHTTP클래스를 호출하여 서버의 ASP 파일에 전송한다. [그림 2]은 통계정보를 얻기 위해 클라이언트에서 서버로 전화번호 정보를 전달하는 기능이다. 서버에서 정보를 가져올 시에는 cmsHTTP 클래스를 호출하여 ASP 파일을 통해 XML형태로 리스트를 받아온다. 받아온 XML데이터는 Util 클래스를 통해 HashMap데이터로 해독하고 출력한다[10]. [그림 3]은 서버로부터 얻은 통계 정보를 텍스트뷰를 통해 출력하는 기능이다.

```
String Url = "http://203.153.147.113/statistics_list.asp";
ArrayList<NameValuePair> httpParams =
    new ArrayList<NameValuePair>();
httpParams.add(new BasicNameValuePair("tel", tel));

cmsHTTP cmsHttp = new cmsHTTP();
cmsHttp.encoding = encoding;
cmsHttp.act = act;

// POST 방식의 웹통신을 실행하여 서버로 정보를 전달
String tmpData = cmsHttp.sendPost(theUrl, httpParams);
if (tmpData==null) return;
Log.i(TAG, tmpData);
```

[그림 2] 클라이언트에서 서버로 정보를 전달하는 기능

```
// 서버로부터 정보를 전달 받음
Util cmsutil = new Util();
hm = cmsutil.xml2HashMap(tmpData, encoding);

listCount = cmsutil.str2int(hm.get("count"));

for(int i=0; i<listCount; i++) {
    stat_roomName.setText(hm.get("roomName["+position+"]"));
    stat_count.setText(hm.get("count["+position+"]"));
}
```

[그림 3] 서버로부터 얻은 정보를 출력하는 기능

### 3.4 알고리즘 분석

본 절에서는 본 시스템을 이용하면서 동작하는 기능 중 채팅방과 관련된 알고리즘을 설명한다.

[그림 4]는 사용자가 채팅방을 생성하는 동작을 설명하는 알고리즘이다. 사용자가 입력한 채팅방 정보가 Input이 되며 community 테이블에 저장되고 com\_id를 리턴한다. 취소를 누르면 Step에서 빠져나간다. 채팅방이 생성되어 입장하면 community 테이블의 채팅방 정보와 loginlist 테이블의 입장한 사용자 정보를 가져와 화면에 출력한다. 출력되는 것은 채팅방명, 현재인원수, 총인원수, 입장한 사용자 리스트가 있다.

#### # Algorithm for Create Chatroom

Input: Chatroom info C

Output: com\_id

#### Algorithm Start:

##### Step 1. 채팅방 생성

채팅방 정보를 community 테이블에 저장

if(취소) exit

Return id

##### Step 2. 채팅방 입장

채팅방 정보를 community 테이블에서 가져와 출력

입장한 사용자 정보를 loginlist 테이블에서 가져와 출력

if(EXIT) exit

Return Message

END

[그림 4] 채팅방 생성 알고리즘

[그림 5]는 사용자가 생성된 채팅방에 입장하고 그 이후 이루어지는 실질적인 대화 과정을 설명하는 알고리즘이다. 입장에 성공하면 현재 채팅방의 상태를 확인한다. 상태는 다른 사용자가 말하고 있는 Wait 상태, 사용자가 말한 내용이 재생 중인 Play 상태 그리고 말하기가 가능한 Talk 상태가 있다. 채팅방의 상태가 Talk이면 말하기 버튼을

눌러 10초 동안 말할 수 있으며 record 테이블에 대화자를 저장한다. 10초가 넘으면 말하기를 중지시켜 발언권을 보호한다. Stop을 누르면 record 테이블에 대화 파일의 제목을 추가하며 대화자와 대화 파일의 제목, 대화 시간을 리턴하고 재생을 시작한다. EXIT를 누르면 Step에서 빠져나간다.

**# Algorithm for Voice Communication**

**Input:** User Name N, Talk Title T, Talk Time P,  
Room State S  
**Output:** id

**Algorithm Start:**

**Step 1.** 채팅방 상태 확인  
record 테이블에서 상태정보를 가져와 화면에 적용  
if(EXIT) exit  
Return Message

**Step 2.** 말하기  
N, S를 record 테이블에 저장  
(P가 10초를 넘으면 말하기 자동 중지)  
if(EXIT) exit  
Return N

**Step 3.** 정지  
T, S를 record 테이블에 추가  
if(EXIT) exit  
Return T, P

**Step 4.** 듣기  
T 정보를 받아서 재생  
if(EXIT) exit  
Return Message

**END**

[그림 5] 보이스 채팅 알고리즘

**4. 구현 결과**

[그림 6](a)는 시스템이 구동되었을 때 나타나는 메인 화면이다. [그림 6](b)는 메뉴 선택 화면으로 버튼을 클릭하면 해당 페이지로 이동한다. [그림 6](c)는 채팅방 정보를 입력하여 채팅방을 생성하는 화면이다. [그림 6](d)는 채팅방에 입장하면 출력되는 채팅방 화면이다.



[그림 6] 결과 화면

**5. 결론**

본 논문은 관심사가 같은 여러 사용자들과 음성으로 대화를 나눌 수 있는 보이스 커뮤니케이션 시스템을 제안하였다. 현존하는 모바일 어플리케이션의 부족한 부분을 개선하여 모바일에서 회원 가입부터 정보 수정 및 탈퇴, 채팅방 생성 및 폐쇄, 그리고 보이스 채팅 기능을 구현하여 그 수요가 많을 것이라 예상된다. 향후 연구로는 사용자의 위치정보를 가지고 GPS를 이용하여 일정한 범위 내의 사용자들과만 커뮤니케이션이 가능하도록 하는 것이다.

**참고문헌**

- [1] 박 현길, “2010년 한국 IT시장은 어플리케이션 마켓,” 한국마케팅연구원 마케팅, 제 45권, 1호, pp.62-70, 2011.1.
- [2] 개인 블로그, <http://blog.naver.com/somangbyun?Redirect=Log&logNo=10098851743>
- [3] 개인 블로그, <http://bment.net/81>
- [4] 개인 블로그, <http://www.cyworld.com/woojiya/3900838>
- [5] 개인 블로그, <http://zayuboy.blog.me/100124392>
- [6] iTunes 홈페이지, <http://itunes.apple.com/kr/app/tikl-touch-to-talk/id396612714>
- [7] iTunes 홈페이지, <http://itunes.apple.com/kr/app/ippt/id311931218?mt=8>
- [8] 최재영·최종명·유재우 공저, “프로그래머를 위한 JAVA2,” 홍릉과학출판사, 2003
- [9] 김상형, “안드로이드 프로그래밍 정복,” 한빛미디어, 2010
- [10] 이두진, “안드로이드 앱 개발 완벽 가이드,” 피씨북, 2010