

모바일 감지 서비스의 신뢰성 향상을 위한 동적 인지 기법 연구 및 개발

은윤규¹, 김철진^{*}

¹인하공업전문대학 컴퓨터시스템과

A Research and Development of Dynamic Recognition Technique for Enhancing Reliability of Mobile Sensing Service

Yun-Kyu Eun¹, Chul-Jin Kim^{*}

¹Dept. of Computer Systems and Engineering, Inha Technical College

요 약 스마트폰은 일상생활의 필수 요소가 되었으며, 스마트폰의 내장된 센서는 사용자의 상황인지를 위해 유용하게 활용될 수 있다. 하지만, 동적 상황 인지를 통해 안전 및 사고예방을 위한 연구가 미흡한 상황이다. 본 논문에서는 모바일 디바이스의 가속도 센서, 마이크, GPS 센서를 이용해 위험상황을 동적으로 인지할 수 있는 기법을 제안한다. 본 동적인지기법의 센싱 기법에 대한 아키텍처와 프로세스를 제안하며, 아키텍처의 적합성을 검증을 위해 모바일 어플리케이션을 개발한다.

Abstract Smartphone has become an essential element in our daily life and built-in sensors of the smartphone can be utilized in order to recognize of user's situation. However, it is lack of research for safety and accident prevention by dynamic situation recognition. In this paper, we propose a technique that can be recognized risk situation dynamically using accelerometer, microphone and GPS sensor of mobile device. We propose an architecture and process for sensing techniques of Dynamic Recognition Technique, and develop the mobile application for verifying the suitability of the architecture.

Key Words : Dynamic Recognition Technique, Mobile Sensing Service, Customization

1. 서론

스마트폰의 사용이 증가함에 따라 많은 사용자들은 스마트폰으로 일상을 시작하고 마감한다. 스마트폰은 단순 전화, 문자 등 통신 기능만을 사용하는 것이 아닌 다양한 분야의 도구로 사용되고 있으며, 스마트폰의 센서를 이용해 사용자 행동을 분석하여 제공하는 어플리케이션의 개발이 매우 활발히 진행되고 있다. 스마트폰을 활용한 상황인지는 다양한 센서를 이용하지만 대부분의 어플리케이션은 단일 센서만으로 개발된다. 하지만 단일 센서를 통한 상황인지의 경우 실생활에 즉각적인 대처를 하기 어려우며, 자동으로 상황을 인지하고 동작하는 어

플리케이션은 부족한 상황이다.

본 논문에서는 실생활에 발생할 수 있는 위험상황을 스마트폰의 가속도 센서, 마이크, GPS 등 다양한 센서를 이용한 동적 인지 기법에 대해 제안한다. 다양한 센서를 통해 감지된 데이터를 분석하여 현 상황을 분석하고, 위험상황으로 인지 될 경우 사용자가 전송하지 않아도 자동으로 위험 정보를 전송한다. 이렇게 다중 센싱을 이용한 동적 인지 기법의 경우 단일 센싱을 통한 상황인지보다 사용자에게 신뢰성 및 편의성을 제공한다.

본 논문은 다음과 같이 구성한다. 2장에서는 관련 연구로 모바일 감지 센서기술 및 모바일 센서 기반 상황인식 연구 사례에 대해 분석하고, 3장에서는 모바일 센서

^{*}Corresponding Author : Chul-Jin Kim(Inha Technical College)

Tel : +82-032-870-2338 email : cjkim@inhac.ac.kr

Received February 11, 2015

Revised (1st March 10, 2015, 2nd April 1, 2015)

Accepted May 7, 2015

Published May 31, 2015

를 활용한 동적 인지 기법을 제안한다. 4장에서는 본 논문에서 제안한 기법을 다중 센서를 이용한 위험감지 어플리케이션에 적용하여 적합성을 검증하고, 5장에서 결론을 맺는다.

2. 관련연구

2.1 모바일 감지 센서기술

스마트폰의 대중화가 이루어져 그에 따라 스마트폰의 내장센서를 이용해 사용자의 행동을 감지하는 다양한 연구가 수행된다. 현재 모바일 감지 센서기술은 가속도 센서, 근접 센서 등 단일 센서를 이용해 사용자의 상황정보를 분석하여 제공한다.

2.1.1 미디어레코더(MediaRecorder)

미디어레코더는 android.media 패키지에 포함되는 클래스로 음성이나 동영상을 기록하기 위한 API(Application Programming Interface)를 제공한다[1].

음성을 녹음 하기위해 Table 1과 같이 지정된 흐름에 따라 설정이 필요하다. 틀린 순서로 설정이 실행되는 경우 오류가 발생 할 수 있다. 음성을 기록하기 위해 Table 2와 같이 prepare() 메소드를 호출 후 start() 메소드를 순차적으로 호출해야하며, 녹음을 종료 할 경우 stop() 메소드 호출 후 release() 메소드가 호출된 후 미디어레코더 인스턴스를 소멸시킨다.

Table 1. Set-up of MediaRecorder

| | |
|-------------------|---|
| setAudioSource() | Sets the audio source to be used for recording. |
| setOutputFormat() | Sets the format of the output file produced during recording. |
| setAudioEncoder() | Sets the audio encoder to be used for recording. |
| setOutputFile() | Sets the path of the output file to be produced. |

Table 2. Functions of MediaRecorder

| | |
|-------------------|---|
| prepare() | Prepares the recorder to begin capturing and encoding data. |
| start() | Begins capturing and encoding data to the file specified with setOutputFile(). |
| stop() | Stops recording. |
| release() | Releases resources associated with this MediaRecorder object. |
| getMaxAmplitude() | Returns the maximum absolute amplitude that was sampled since the last call to this method. |

2.1.2 센서관리자(SensorManager)

안드로이드에는 Table 3과 같이 다양한 센서가 적용되어 있다. 센서 관리자는 이러한 다양한 센서를 디바이스에서 사용할 수 있도록 도와주는 역할을 한다[2].

Table 3. Sensors of Android

| | |
|---------------------|---|
| TYPE_ACCELEROMETER | A constant describing an accelerometer sensor type. |
| TYPE_MAGNETIC_FIELD | A constant describing a magnetic field sensor type. |
| TYPE_ORIENTATION | A constant describing an orientation sensor type. |
| TYPE_GYROSCOPE | A constant describing a gyroscope sensor type. |
| TYPE_LIGHT | A constant describing a light sensor type. |
| TYPE_PRESSURE | A constant describing a pressure sensor type. |
| TYPE_TEMPERATURE | A constant describing an ambient temperature sensor type. |
| TYPE_PROXIMITY | A constant describing a proximity sensor type. |
| TYPE_ALL | A constant describing all sensor types. |

센서는 독립된 하드웨어 장비이므로 부팅 직후부터 항상 동작중이다. 센서 사용을 위해 Table 4의 getSystemService() 메소드를 통해 디바이스의 센서 인스턴스를 얻어올 수 있다. 사용자가 원하는 센서는 getDefaultSensor() 메소드를 통해 권한을 얻어올 수 있으며, SensorEventListener 인스턴스의 registerListener() 메소드를 통해 사용자가 원하는 센서와 함께 등록한다. 센서의 값은 끊임없이 변하며 이때마다 등록된 리스너의 onSensorChanged() 메소드를 통해 값을 전달한다.

Table 4. Functions of SensorManger

| | |
|--------------------|---|
| getSystemService() | Return the handle to a system-level service by name. |
| getDefaultSensor() | Use this method to get the default sensor for a given type. |
| registerListener() | Registers a SensorEventListener for the given sensor. |
| onSensorChanged() | Called when sensor values have changed. |

2.2 모바일 센서 기반 상황인식 연구

2.2.1 스마트폰 가속도 센서 기반의 돌발상황인식 어플리케이션 개발

연구 [3]에서는 가속도 센서 기반의 돌발상황을 자동으로 감지하고 이를 사용자에게 즉시 알려주는 기법을

제시한다. 어플리케이션이 실행되면 가속도 데이터를 누적하여 누적된 데이터의 평균값을 계산하여 평균값을 벗어날 경우 디바이스를 통해 경보를 울려 사용자에게 알려줄 수 있다.

이는 돌발상황(도난, 추락 등) 발생 시 가속도 센서와 근접 센서 사용을 이용해 단일 사용자에게만 경보가 발생한다. 이는 본 논문에서 제안한 동적 인지 기법이 적용된 어플리케이션은 위험상황 시 가속도 센서와 음성 인식, 위치 인식 등 다중 센싱이 적용되어 보다 향상된 위험상황인지가 제공된다.

2.2.2 모바일 헬스케어 지원을 위한 스마트폰을 이용한 낙상 감지 시스템

연구 [4]에서는 가속도 센서와 GPS 센서 기반의 낙상감지 시스템을 제안하였다. 가속도 센서 정보를 이용하여 자세를 예측하고 그에 따라 사용자가 낙상이 발생하였을 경우 GPS 센서를 통해 발생한 위치정보를 서버로 전달하여 저장 후 모니터링 디바이스로 경보를 발생시켜 도움을 요청한다.

가속도 센서를 통해 움직임을 감지하나 낙상감지 시 적절히 반응하지 않을 경우 위험을 효과적으로 알릴 수 없으며 서버가 항상 구동되어야 하는 단점이 존재한다. 본 논문에서 소개한 기법을 적용 할 시 다중 센싱을 사용해 단일 센싱 기법보다 향상된 인지가 가능하다.

2.2.3 스마트폰 내장 가속도 센서를 이용한 2단계 행위 인식 시스템의 설계 및 구현

연구 [5]에서는 가속도 센서를 이용한 2단계 행위 인식 시스템을 제안하였다. 가속도 센서 정보를 이용하여 사용자의 서기, 앉기, 걷기, 뛰기 등 행위를 인식하고 데이터 수집 및 전처리, 특징 추출, 모델 학습 및 분류를 통해 보다 향상된 인지를 제공한다.

단일 센서를 이용하여 행위를 인식하기 때문에 본 논문에서 제안한 다중 센서를 이용한 동적 인지 기법을 통해 보다 적절한 대응이 가능하다.

2.2.4 패미

어플리케이션 [6]에서는 GPS 센서 기반의 위치추적 어플리케이션을 개발하였다. GPS 센서를 통해 그룹원들의 위치를 실시간으로 확인이 가능하며 1주일간의 이동 기록을 저장한다. 스마트폰 분실 시 저장된 위치정보를 통해 스마트폰의 위치 확인이 가능하다.

위험이 발생할 경우 사용자가 수동적으로 SOS 버튼을 클릭하는 번거로움이 존재한다. 본 논문에서는 동적 상황인지를 통해 연구 [6] 보다 사용자의 편의성을 향상시킬 수 있다.

3. 동적 인지 기법

본 연구는 모바일 디바이스의 감지 센서를 이용하여 동적으로 인지할 수 있는 기법을 제안하며 모바일 감지 서비스의 신뢰성을 향상시킬 수 있을 것이다.

3.1 동적인지 아키텍처

본 논문에서 제안하는 동적 인지 기법을 위한 구조는 Fig. 1과 같이 어플리케이션 계층(Application Layer), 푸시 서비스 계층(Push Service Layer), 인식 계층(Sensing Layer), 하드웨어 계층(Hardware Layer)으로 구성된다.

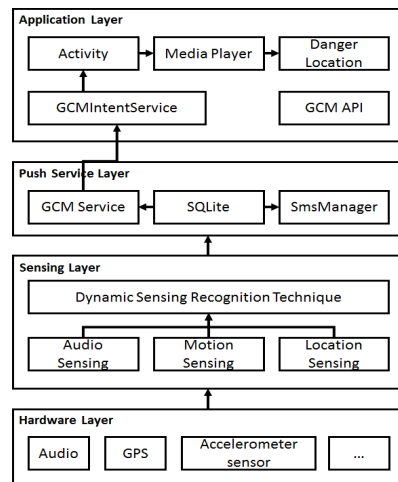


Fig. 1. Dynamic Sensing Architecture

사용자의 디바이스는 항상 인식 계층을 통하여 음성 인식(Audio Sensing)과 동작 인식(Motion Sensing), 사용자의 현재 위치를 확인하기 위한 위치 인식(Location Sensing)을 제공한다. 음성 인식이나 동작 인식과 같이 다중의 인식 방법을 이용해 상호보완적 상황인식 기능을 제공하여 신뢰성을 향상시킬 수 있다.

디바이스는 사용자의 행동을 감지하여 위험여부를 판단하며, 음성 인식의 경우 하드웨어의 음성을 통해 입력

된 음성데이터를 데시벨(decibel, dB)화 하여 지정한 음성이 기준 값을 초과 하였을 경우, 또한 동작 인식의 경우 가속도 센서(Accelerometer Sensor)를 통해 3개의 z, x, y 축 값이 급변할 경우 위험으로 판단한다. Fig. 2와 같이 사용자 상황정보를 입력 받아 커스터마이제이션 모듈을 통해 사용자의 행동을 동적으로 확인하여 음성이 인식되지 않거나 주변소음이 심할 경우에는 음성인식을 일시 정지 후 동작 인식을 통해 감지하여 위험을 알리며 반대의 경우도 마찬가지이다.

위험 상황 시 음성 인식과 동작 인식의 상호보완을 통해 디바이스 감지의 신뢰성을 향상 시킬 수 있다.

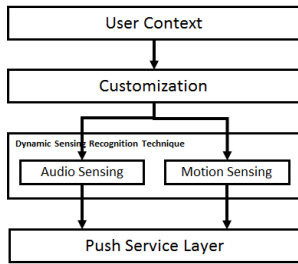


Fig. 2. Sensing Customization Architecture

위험이 인식된 이후에는 푸시 서비스 계층의 GCM(Google Cloud Messaging) 서비스[7]와 SMS 관리자[8]를 통해 위험 정보를 전송한다. GCM 서비스와 SMS 관리자는 본인과 그룹으로 구성된 참여자들에게 위험 상황을 전송한다. 이때 위험이 발생한 위치 및 시간이 포함된 정보를 전송한다. GCM 서비스를 통해 전송된 정보는 GCM 서버를 통해 GCMIntentService로 수신된다. 수신된 데이터를 통해 사용자의 위험정보를 나타내며, 미디어플레이어(MediaPlayer)를 통해 강제적으로 위험신호를 나타낸다.

3.2 동적 인식 프로세스

동적 인식 프로세스는 백그라운드 프로세스(Background process)로 실행되며, 미디어 레코더를 통한 음성 인식과 센서 관리자를 통한 동작 인식으로 위험 상황을 감지한다.

음성 감지를 위한 프로세스는 다음과 같다. 음성 인식을 위해 안드로이드 모바일 플랫폼에서 미디어 레코드 허용할 수 있도록 AndroidManifest.xml을 설정하여야 한다(Fig. 3).

```

    <uses-permission
        android:name="android.permission.RECORD_AUDIO" />
  
```

Fig. 3. Permission Code for MediaRecorder

미디어 레코더를 설정 한 후 원하는 음성 데이터를 입력받기 위하여 미디어 레코더의 setAudioSource(), setOutputFormat(), setAudioEncoder(), setOutputFile() 메소드를 통해 입력, 출력, 인코더, 출력 파일 형식을 설정해야 한다(Fig. 4).

```

    mRecorder = new MediaRecorder();
    mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
    mRecorder.setOutputFile("/dev/null");
  
```

Fig. 4. Input Code for Audio Data

이와 같이 사용자의 음성 데이터를 입력받기 위해 설정이 필요하며, 입력 받은 음성데이터를 데시벨 화 하기 위해 미디어 레코더의 getMaxAmplitude() 메소드를 통해 입력된 최대 진폭을 리턴 받아 데시벨을 계산 한다. 처리된 데시벨 값이 기준 값을 초과 하였을 경우 위험 상황으로 인지한다. 기준 값에 도달 하지 못 할 경우 주기적으로 음성 데이터를 입력 받아 위험상황에 대비한다 (Fig. 5).

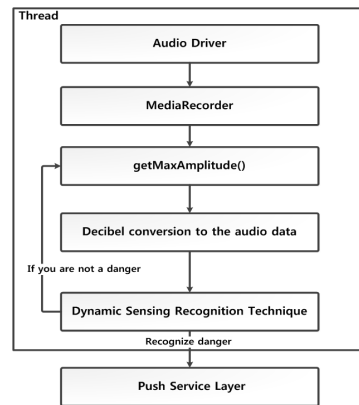


Fig. 5. Dynamic Audio-Sensing Process

동작 감지를 위한 프로세스는 센서 관리자의 `getSystemService()` 메소드를 통해 센서 관리자 인스턴스를 얻어오며, 센서 관리자 인스턴스의 `getDefaultSensor()` 메소드를 통해 가속도 센서 사용을 위한 권한을 획득한다. 센서 관리자를 이용하여 상속받은 `SensorEventListener` 클래스를 리스너에 등록한다 (Fig. 6).

```

public void sensorInit() {
    sensorManager = (SensorManager)
        getSystemService(SENSOR_SERVICE);
    sensor =
        sensorManager.getDefaultSensor(
            Sensor.TYPE_ACCELEROMETER);
    sensorEventListener = new accListener();
    sensorManager.registerListener(sensorEventListener,
        sensor, SensorManager.SENSOR_DELAY_NORMAL);
}
    
```

Fig. 6. Input Code for Accelerometer Data

Fig. 7의 흐름을 통해 등록된 리스너는 디바이스 센서의 데이터가 변경 될 경우 `onSensorChanged()` 메소드를 호출하여 디바이스의 흔들림을 감지한다. 기준 값에 도달 하지 못 할 경우 주기적으로 센서 데이터를 입력 받아 위험상황에 대비한다.

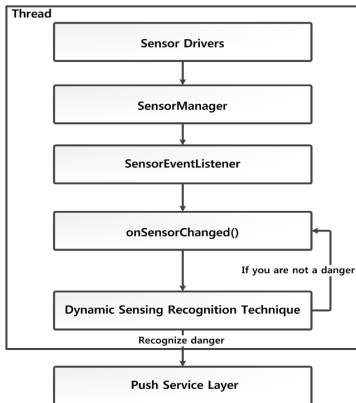


Fig. 7. Dynamic Motion-Sensing Process

사용자의 위치를 동적으로 확인하기 위해 Fig. 8과 같은 흐름을 거쳐 위치 관리자(Location Manager)에게 위치 서비스를 요청한다[9]. GoogleMap을 사용하기 위한 코드는 Fig. 9와 같다. 사용자의 현재 위치를 나타내기

위해 Layout과 GoogleMap을 연결시키며 `setMyLocationEnabled()` 메소드를 통해 사용자의 위치를 표시한다. `setOnMyLocationChangeListener()` 메소드를 통해 사용자의 위치가 변경될 때 마다 `onMyLocationChange()` 메소드가 호출된다.

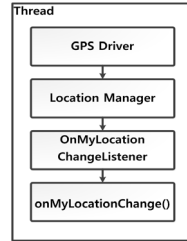


Fig. 8. Data Flow for Dynamic Location Service

```

LocationManager =
    (LocationManager) getSystemService
    (Context.LOCATION_SERVICE);
GoogleMap =
    ((MapFragment) getFragmentManager())
    .findFragmentById(R.id.map).getMap();
GoogleMap.setMyLocationEnabled(true);
GoogleMap.setOnMyLocationChangeListener(this);
    
```

Fig. 9. Input Code for Location Data

3.3 동적 인식 메시지 프로세스

인식 계층을 통해 위험상황으로 인지 될 경우 Fig. 10과 같은 흐름을 통해 메시지가 송신된다. 또한 사용자의 수동적인 위험 발생 이벤트에 대해서도 동일한 메시지 흐름을 갖는다.

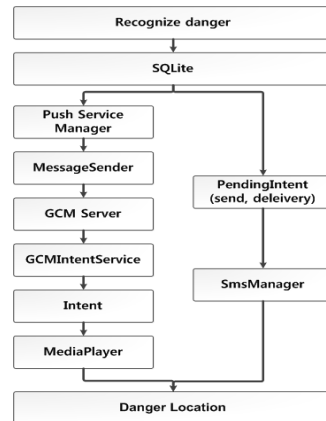


Fig. 10. Message Flow for Dynamic Sensing

위험상황정보를 전송하기 위해 데이터베이스(SQLite)[10]에 저장된 그룹정보를 조회하여 해당 사용자들에게 전송한다. 전송 시 GCM 서비스를 이용하기 위해 AndroidManifest.xml 파일에 퍼미션을 설정하여야 한다(Fig. 11).

```
<permission
    android:name="com.guardianangel.permission.C2D_MESS
    AGE"
    android:protectionLevel="signature" />
<uses-permission android:name="com.guardianangel.perm
    ission.C2D_MESSAGE" />
<uses-permission android:name="com.google.android.c2d
    m.permission.RECEIVE" />
<uses-permission android:name="android.permission.GET_
    ACCOUNTS" />
<uses-permission android:name="android.permission.WAK
    E_LOCK" />
<uses-permission android:name="android.permission.INTE
    RNET" />
```

Fig. 11. Permission Code for GCM

데이터베이스에 기록된 등록 ID(Registration ID)를 이용해 GCM Message Sender를 통하여 데이터를 전달한다. 이 데이터를 받은 GCM Server는 어플리케이션 계층의 GCMIntentService를 통해 데이터를 수신하게 된다. 위험상황정보 데이터 수신 후 Intent를 통해 새로운 Activity를 생성하고 Activity를 통해 애니메이션과 미디어 플레이어를 강제적으로 발생시키며, 해당 Activity를 클릭하여 위험이 발생한 위치를 확인할 수 있다.

SmsManager를 이용하기 위해 AndroidManifest.xml 파일에 퍼미션 설정을 한다(Fig. 12).

```
<uses-permission
    android:name="android.permission.SEND_SMS" />
```

Fig. 12. Permission Code for SMS

데이터베이스에 저장된 정보를 바탕으로 SMS의 송신 및 수신 성공, 실패 등의 이벤트 사용을 위해 PendingIntent가 필요하며, SmsManager의 sendMessage() 메소드를 통해 송신한다(Fig. 13).

```
PendingIntent sentIntent =
    PendingIntent.getBroadcast(this, 0,
        new Intent(SEND_INTENT), 0);
PendingIntent deliveryIntent =
    PendingIntent.getBroadcast(this, 0,
        new Intent(DELEIVERY_INTENT), 0);
try {
    Cursor cursorPhone =
        mDB.selectAll(DBAdapter.GROUP_TABLE_NAME);
    if (cursorPhone != null && cursorPhone.getCount() != 0)
    {
        cursorPhone.moveToFirst();
        SmsManager sms = SmsManager.getDefault();
        do {
            sms.sendMessage(cursorPhone.getString(2), null,
                "SOS!#nhhttp://maps.google.com/maps?f=q&q=(" +
                Latitude + "," + Longitude + ")&z=18", sentIntent,
                deliveryIntent);
        } while (cursorPhone.moveToNext());
    } else { }
    cursorPhone.close();
} catch (Exception ex) { }
```

Fig. 13. SMS Message Code

4. 실험 및 평가

본 논문에서는 모바일 감지 서비스의 신뢰성 향상을 위한 동적 인지 기법을 제안하였다. 이에 대해 사용자의 위험 상황을 다중 센서를 기반으로 동적 인식하는 어플리케이션을 통하여 적합성을 검증한다. 본 어플리케이션은 사용자 등록 기능, 위험상황을 감지하는 기능, GCM을 통해 위험상황 정보를 알리는 기능으로 구성된다.

4.1 사용자 등록

사용자 등록 기능은 사용자에게 등록 기능에 대한 편리성을 제공하기 위해 NFC(Android Beam)[11]와 GCM을 이용하여 구현한다. 등록하고자 하는 두 사용자의 디바이스를 서로 맞대어 송신과 수신 대기 상태에서 등록이 가능하다. 등록 시 전송되는 데이터는 해당 디바이스의 정보와 GCM 사용을 위한 등록 ID를 수신 상태인 디바이스로 전송한다. 수신 받은 디바이스에서는 전송 받은 디바이스 정보와 등록 ID를 데이터베이스에 저장 후, 전송 받은 등록 ID를 이용하여 본인의 디바이스 정보를 GCM을 통해 상대 디바이스에 전송하여 등록할 수 있도록 한다(Fig. 14). NFC의 경우 양방향 통신이 가능하지

않기 때문에 이를 보완해 GCM을 통해 두 사용자간에 상호 등록이 가능하다.



Fig. 14. User Registration using NFC

Fig. 15는 NFC와 GCM을 이용하여 안드로이드 기반의 디바이스에서 두 사용자들 간에 등록하는 사례이다.

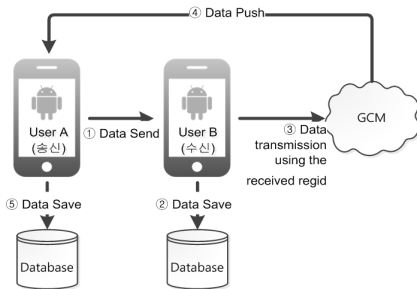


Fig. 15. User Registration Architecture

4.2 위험상황 동적인식

위험 상황인식은 디바이스의 백그라운드 프로세스로 실행되며, 미디어 레코드를 통해 음성 인식과 동작 인식을 통하여 위험상황을 동적으로 감지한다. 기존 어플리케이션들은 사용자가 직접 위험상황에 대해 수동적으로 반응해야 하는 것과 비교하여 본 연구의 경우 사용자가 수동적으로 동작할 필요 없이 위험상황에 대해 자동적으로 감지할 수 있다.

동작 인식의 경우 디바이스의 흔들림을 통해 가속도 센서의 z, y, x 축의 변화를 통하여 위험상황을 판단하며 음성 인식의 경우 소리의 데시벨을 계산하여 85 ~ 95 데시벨 이상 발생 시 위험상황으로 판단한다. 만약 위험상황 발생 시 음성 인식이 작동하지 않을 경우 디바이스의 흔들림을 통해 위험상황을 동적으로 인식하여 위험상황을 통보한다. 반대로 동작 인식이 작동하지 않을 경

우 음성 인식을 작동하여 위험 상황을 통보한다. 위험 상황인식의 신뢰성 향상을 위하여 위험상황이 중첩(예, 음성감지 2회, 동작감지 3회)될 경우 위험정보를 GCM을 통하여 통보하게 된다.

4.3 위험상황정보 전송

디바이스의 동적 센싱을 통하여 위험상황이 감지 될 경우 데이터베이스에 저장된 정보를 조회하여 GCM 서비스와 SMS를 통해 전송한다. 위험감지가 될 경우 위치관리자를 통해 경도와 위도를 조회 하여 위험상황이 발생한 위치정보와 시간을 함께 전송한다. 이를 통해 위험이 발생한 위치를 확인 할 수 있다.

위험상황발생시 화면 흐름은 다음 Fig. 16과 같다. GCM을 통해 수신된 메시지를 위험상황으로 간주 될 경우 강제적으로 경보와 애니메이션을 발생시키며 또한 위험상황정보를 기록한다. 위험상황정보는 화면을 클릭하여 넘어갈 수 있다. 기록 된 로그를 통해 위험이 발생한 위치를 확인 할 수 있으며, SMS로 수신 된 메시지 또한 동일하다.

GCM을 통해 수신된 데이터는 위험이 감지된 위치와 발생한 시간을 데이터베이스에 저장한다. 위험정보를 선택하여 위험이 발생한 위치를 확인 할 수 있다(Fig. 17).

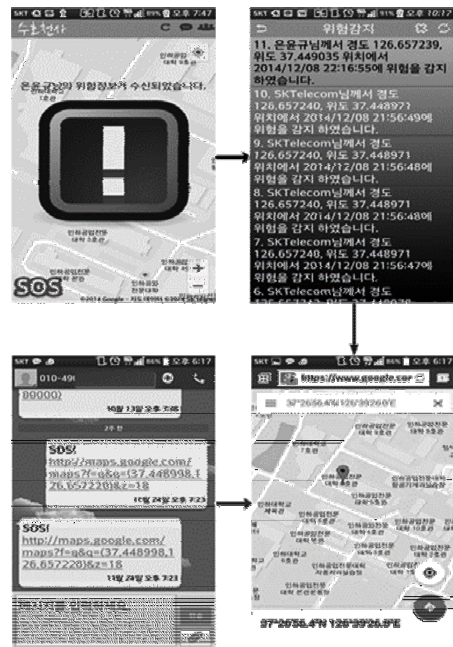


Fig. 16. Dynamic Sensing Process

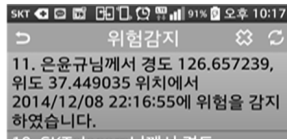


Fig. 17. Notification Message

SMS를 통해 전송되는 정보는 GoogleMap을 통해 바로 확인이 가능하도록 위험감지 위치가 포함되어 전송된다(Fig. 18).

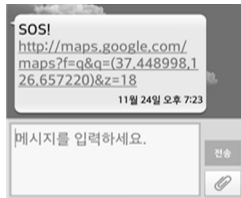


Fig. 18. Notification Message including Location

기존 단일 센싱만으로 적용된 어플리케이션보다 본 논문을 통해 제안된 동적 인지 기법과 커스터마이제이션이 적용된 어플리케이션은 다중 센서들 간에 상호보완을 통해 하나의 센서가 정상동작 하지 않을 경우 다른 센서를 이용해 위험을 감지함으로써 신뢰도를 높일 수 있다.

5. 결론

본 연구에서는 모바일 디바이스의 감지 센서를 이용하여 동적으로 인지할 수 있는 기법을 제안하였다. 가속도 센서와 마이크, GPS를 이용해 지속적인 센서 데이터를 분석하고 사용자의 위험상황을 즉각적으로 감지하여 등록된 사용자에게 위험상황을 알리기 위한 기법을 제안하였다. 이는 단일 센서만을 적용한 어플리케이션보다 다중 센싱을 통해 분석 할 경우 하나의 센서가 감지하지 못 할 경우 다른 센서를 이용해 위험을 감지함으로써 보다 신뢰성 있는 상황인지가 가능하다.

향후 연구는 사용자의 일상적인 행동 패턴(예, 일정 거리 등)을 분석하여 행동 패턴을 벗어날 경우 위험을 감지하도록 하는 기술을 연구한다.

References

- [1] Android Developers, "MediaRecorder", <http://developer.android.com/reference/android/media/MediaRecorder.html>, 2015
- [2] Android Developers, "SensorManager", <http://developer.android.com/reference/android/hardware/SensorManager.html>, 2015
- [3] Kyung-Ae Cha, SunDong Yeo, "Smart phone Application Development for Aware of Unexpected Conditions using Accelerometer Sensors", 2012
- [4] Pil-Seong Jeong, Yang-Hyun Cho, "Fall Detection System using Smartphone for Mobile Healthcare", p: 435-447, 2013
- [5] Jong-Hwan Kim, In-Cheol Kim, "Design and Implementation of a Two-Phase Activity Recognition System Using Smartphone's Accelerometers", 2014
- [6] GooglePlay, "FAMY", <https://play.google.com/store/apps/details?id=com.spacosa.android.famy.global>, 2015
- [7] Tistory, "Using GCM 1/3", <http://leminity.tistory.com/26>, 2013
- [8] Android Developers, "SmsManager", <http://developer.android.com/reference/android/telephony/gsm/SmsManager.html>, 2015
- [9] Android Developers, "LocationManager", <http://developer.android.com/reference/android/location/LocationManager.html>, 2015
- [10] Android Developers, "SQLiteDatabase", <http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html>, 2015
- [11] Tistory, "What is NFC?". <http://big.tistory.com/77>, 2014

은 윤 규(Yun-Kyu Eun)

[준회원]



• 2008년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 재학

<관심분야>

소프트웨어 공학, 모바일 서비스, 데이터베이스

김 철 진(Chul-Jin Kim)

[종신회원]



- 2004년 2월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)
- 2004년 3월 ~ 2009년 2월 : 삼성 전자 책임연구원
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 부교수

<관심분야>

컴포넌트 기반 개발 방법론, 컴포넌트 커스터마이제이션, 모바일 서비스, 클라우드 컴퓨팅