

모바일용 마인드 맵 노트 구현

윤정원*, 임문규*, 배성호*, 계주성*, 김우생*

*광운대학교 컴퓨터소프트웨어학과

e-mail : kyejusung@cs.kw.ac.kr

Implementation Mind-map note on Mobile

Jungwon Yoon*, Mungyu Lim*, Sungho Bae*, Jusung Kye*, Woosaeng Kim*

*Dept of Computer Software, Kwang-Woon University

요 약

기존의 휴대폰에 탑재된 메모기능은 단순히 문자정보만 기록할 수 있기 때문에 실용성과 효율성이 낮다. 이를 보완하기 위해 본 연구에서는 마인드맵(Mindmap)표기법을 도입하여 메모 간의 연계성을 높이고 관리를 용이하게 하며 다양한 멀티미디어 자료를 포함할 수 있도록 하였다. 그리고 이러한 기능들을 구글의 개방형 모바일 플랫폼인 안드로이드(Android)에서 구현하였다.

1. 서론

현대사회에서 사람과 휴대폰은 불가분의 관계가 되었다. 단순히 휴대폰을 이용해 통화하고 메시지를 주고받는 기능 이외에도 음악 감상, TV 및 동영상 시청, 게임, 스케줄 관리, 사진, 디지털카메라 등 많은 용도로 휴대폰을 사용하고 있다. 휴대폰의 부가 기능 중에서 메모장 기능은 종이나 펜이 없이도 꼭 기억해야 할 사항이거나, 즉흥적으로 떠오르는 아이디어를 휴대폰에 저장할 수 있게 해준다. 하지만 이 기능은 단순히 텍스트 정보를 나열하는 것이기 때문에 떠오르는 생각들을 언제라도 필요할 때 즉시 찾아내어 쓸 수 있도록 조직화하여 정리하는 것이 불가능하며 다양한 종류의 멀티미디어 자료를 포함하는 것이 불가능하다.

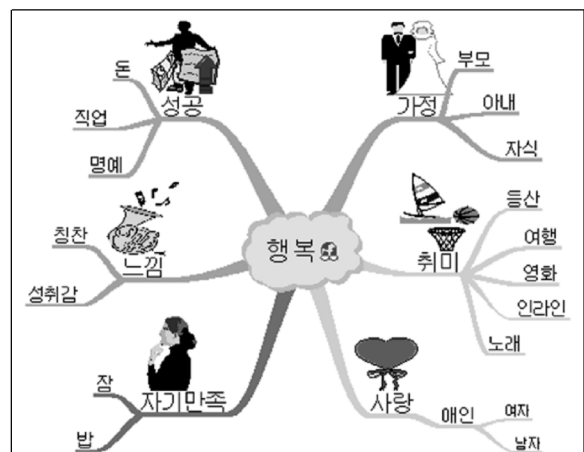
본 연구에서는 이러한 기존 휴대폰 메모기능의 한계를 극복하기 위해 마음속에 지도를 그리듯 글자, 기호, 그림 등을 사용하여 생각을 표현하는 마인드맵(MindMap) 표기법을 적용하였다. 그리고 이것을 구글의 개방형 모바일 플랫폼인 안드로이드(Android)를 이용하여 여기서 제공해주는 SDK를 사용해서 UI(User Interface)를 제작하였고, 마인드맵 형태의 정보를 쉽게 관리하기 위해 XML파일로 저장하여 확장성과 범용성을 고려하였다. 이를 통해 보다 향상된 메모 기능을 할 수 있는 응용프로그램을 구현하였다.

본 논문은 다음과 같이 구성되어있다. 2장에서 마인드맵 노트법과 안드로이드에 대하여 살펴보고 3장에서는 안드로이드의 SDK를 이용하여 마인드맵 노트, 멀티미디어 자료 첨부, 작성된 노트를 저장하고 읽어오기 위한 XML파일 구조를 설계하고 4장에서는 3장에서 설계한 기능 및 UI를 구현한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

2.1 마인드맵(MindMap)

마인드맵은 1971년 영국의 토니 부잔에 의해 창시된 방법으로 두뇌 활용을 극대화 할 수 있는 노트법이다. '중심체로부터 사방으로 뻗어나간다'는 의미를 지닌 방사사고의 표현으로, 마음속에 지도를 그리듯 글자, 기호, 그림 등을 사용하여 생각을 표현하고 인식하여 이미지와 언어의 연상 작용을 효과적으로 활용하는 노트법이라 할 수 있다. 마인드맵 노트기법은 그림 1과 같이 가운데에 노트의 주제가 되는 단어를 두고 그 단어를 중심으로 그와 관련되거나 속하는 범주의 주제나 아이디어를 가지처럼 뻗치게 트리(Tree) 형태로 확장시켜 나간다. 또한 연관되는 주제나 아이디어 끼리 화살표나 나선으로 연결시킬 수도 있다[1].



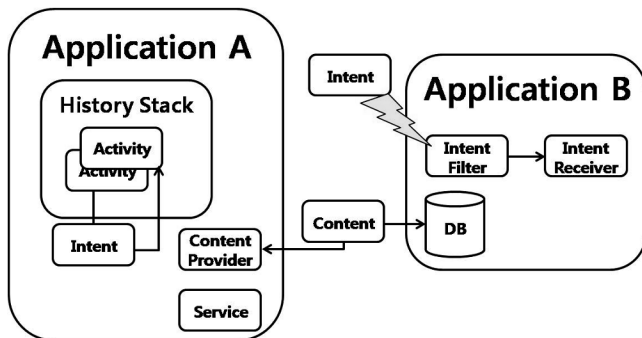
(그림 1) 마인드맵 노트의 예

2.2 안드로이드 응용프로그램

안드로이드는 Google과 OHA(Open Handset Alliance)가 개발한 휴대폰을 위한 소프트웨어 개발 플랫폼이다. OS는 Linux로 개발되었으며, Java를 이용하여 개발을 할 수 있도록 다양한 툴을 제공한다 [2,3]. 안드로이드 플랫폼은 크게 응용프로그램을 구성하는 기능별 컴포넌트에 해당하는 응용프로그램 구성요소와 사용자에게 보이는 것에 해당하는 화면(UI) 구성요소로 나눌 수 있다.

2.2.1 응용프로그램 구성요소

안드로이드 응용프로그램은 그림 2와 같이 Activity, Intent Receiver, Service, Content Provider 4가지의 주요한 클래스로 구성되어 있다[4,5].



(그림 2) 응용프로그램 구성

Activity는 응용프로그램에서 하나의 화면을 지칭한다. 각 Activity는 Activity 클래스를 상속하여 구현한 하나의 클래스이며, 사용자에게 View와 Event 응답으로 이루어진 인터페이스를 화면상에 보여준다. 예를 들어 휴대폰에서 문자를 보낼 때, 첫 화면에서 주소록을 보여 주고, 두 번째 화면에서는 선택한 사람에게 메시지를 쓸 수 있도록 하고, 그다음 화면에서는 보낸 메시지를 확인하거나 환경설정을 바꾸는 화면이라고 가정한다면, 이러한 화면들이 Activity가 되며, 다른 화면으로의 이동은 새로운 Activity를 시작하는 것과 같다. 화면이 오픈되면, 이전 화면은 멈추게 되며, 화면에 대한 정보가 History Stack에 저장되어, 사용자가 원할 때 이전화면으로 돌아갈 수 있도록 한다. 이 History Stack은 응용프로그램 마다 하나씩 존재한다.

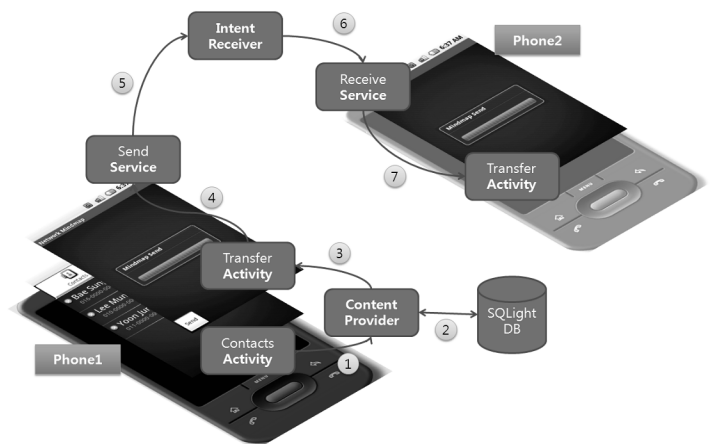
Intent Receiver는 일종의 이벤트 핸들러로써 외부에서 전달받은 Intent를 처리하는 역할을 한다. Intent는 Activity와 Activity를 이동할 때 어떠한 작업에 대한 요청을 담는 클래스이며, 행동에 대한 Action(Activity)과 사용할 데이터(Bundle)로 구성되어 있다. 예를 들어, 어떤 사람의 연락처를 보려 할 때 VIEW Action에 대한 Intent를 생성하고 그 사람

을 표현하는 URI(Uniform Resource Identifier) 데이터를 만들어 VIEW Action의 데이터로 지정해주면 된다. 그 후 Intent Receiver가 외부에서 날아온 Intent를 받게 되면 사용자에게 어떤 일이 발생했다는 것을 알려준다.

Service는 윈도의 서비스나 유닉스의 데몬과 유사하며, 화면상에 보이지 않는 상태로 작업을 수행한다. 사용자는 보통 음악을 계속 들으면서 게임을 즐기거나, 인터넷 서핑을 하기를 원하고 이러한 경우에 일반적인 응용 프로그램을 통해 처리가 불가능하므로 Service를 사용한다.

Content Provider는 구글맵, 전화번호부 같은 모든 콘텐츠(Content)나 리소스(Resource)를 응용 프로그램들끼리 공유할 수 있도록 하고 있으며, 이 콘텐츠들은 안드로이드에 내장된 SQLite DB를 통해 생성되고 관리된다.

항상 모든 응용프로그램이 이 4가지 구성요소를 포함하지는 않는데, 그림 3은 이것들을 모두 포함하고 있는 핸드폰 사이에 주소록을 공유하는 응용프로그램의 예이다. 실제 전송을 담당하는 응용프로그램은 주소록 데이터를 가지고 있지 않지만, Content Provider를 통해 주소록 응용프로그램의 데이터를 가지고 올 수 있으며, 이를 Service와 Intent Receiver를 통해 전송과정을 수행한다.

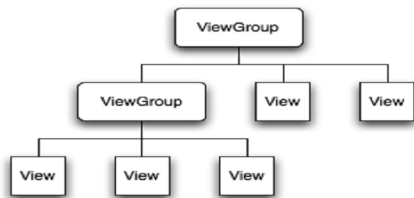


(그림 3) 주소록 공유프로그램의 데이터 전송과정

2.2.2 화면 구성요소

안드로이드의 UI는 사용자 인터페이스를 표현하는 기본단위인 View와 ViewGroup으로 구성되며, 이 두 클래스는 Composite 패턴으로 구현되어 있다. View 클래스는 화면상에서 직사각형 영역의 레이아웃과 내용을 저장하는 자료 구조로써 화면에서 정렬, 그리기, 포커스 이동, 키의 동작 제어를 담당하는 메

소드를 제공한다. ViewGroup은 자신에게 포함된 View와 ViewGroup을 제어하는 기능을 갖는 클래스로써 ViewGroup은 자신이 포함하고 있는 View들을 화면상에 배치하고, 이것들을 하나의 개체처럼 관리할 수 있도록 해준다. 안드로이드에서는 XML 코드를 이용하여 그림 4와 같은 View와 ViewGroup의 구조 관계를 정의함으로써 UI를 디자인한다.



(그림 4) Composite 패턴이 적용된 화면 구성요소

3. 설계

3.1 기능 설계

모바일 환경에서 마인드맵 노트를 구현하기 위해 다음과 같이 기능과 구조를 설계하였다. 노드(Node)를 하나의 마인드맵 노트의 내부에 삽입되는 아이디어(혹은 주제)로 정의하고, 이 노드들을 트리 형태로 구성하여 노드의 추가, 삭제, 이동 기능을 설계하였다. 이 때, 하나의 노드에는 기본적으로 텍스트 정보와 함께 이미지와 멀티미디어 파일을 삽입할 수 있도록 하되 파일을 실제로 노드에 삽입하는 것이 아니라, 휴대폰의 파일시스템에 존재하는 파일의 주소만 참조하도록 설계하였다.

또한, 노트내의 노드의 개수가 많아지게 되면, 상대적으로 크기가 작은 휴대폰 화면은 효과적으로 노트의 개괄적인 모습이나 내용을 보여주는 것이 불가능하다. 따라서 터치패드를 이용하여 화면을 상하 좌우로 이동할 수 있도록 하는 기능과 화면의 비율을 확대하거나 축소하는 기능을 추가로 설계하였으며, 효율적인 노트관리를 위해 자식 노드를 분리해서 새로운 노트로 만드는 분리기능과 노트를 서로 병합하는 기능도 설계하였다.

3.2 마인드맵 파일구조

작성된 마인드맵은 XML 파일로 저장된다. 마인드맵은 하나의 최상위 노드 아래에 다수의 하위 노드를 가진다. 각 노드는 문자열 형식의 제목과 본문을 포함하며, 여러 개의 이미지와 멀티미디어 자료를 가질 수 있다. 이러한 마인드맵을 저장하기 위한 XML파일의 DTD는 그림 5와 같다. map 원소는 마인드맵을 시작하는 최

상위 원소이며, node 원소는 노드 하나를 의미한다. 노드의 제목은 title이라는 이름의 속성값으로 저장되며, 본문은 body 원소로 저장된다. 노드에 포함된 이미지, 멀티미디어 자료는 각각 image, media 원소로 저장되며, src 속성에 자료가 저장된 절대경로를 저장한다.

```

<!DOCTYPE map [
  <!--ELEMENT map (node*)-->
  <!--!ATTLIST map version ID #REQUIRED-->
  <!--ELEMENT node( node*, body, media*, image*)-->
  <!--!ATTLIST node title ID #REQUIRED-->
  <!--ELEMENT body (#PCDATA)-->
  <!--ELEMENT media (#PCDATA)-->
  <!--!ATTLIST media src ID #REQUIRED-->
  <!--ELEMENT image (#PCDATA)-->
  <!--!ATTLIST image src ID #REQUIRED-->
]>
  
```

(그림 5) 저장되는 XML파일의 DTD

4. 구현

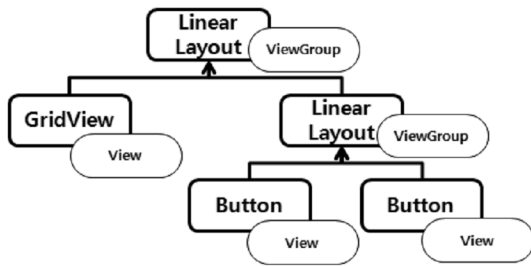
본 연구는 MS Windows XP에 안드로이드 플랫폼(m5-rc15)을 설치하였고, 개발 툴은 Eclipse에 안드로이드 개발 플러그인(adt)을 설치하여 사용하였다.

안드로이드의 UI는 기존의 다른 플랫폼과는 달리 XML을 이용하여 편하게 디자인할 수 있다. 그림 6은 UI 디자인의 예로, 멀티미디어 삽입부의 UI 디자인을 위한 XML코드이며 그림 7과 같은 구조를 가진다.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <GridView android:id="@+id/FileExplorer_layout"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="10"
        android:columnWidth="60"
        android:stretchMode="columnWidth"/>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <Button android:id="@+id/FileExploer_button_ok"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:text="Open"/>
        <Button android:id="@+id/FileExploer_button_cancel"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:text="Cancel"/>
    </LinearLayout>
</LinearLayout>
  
```

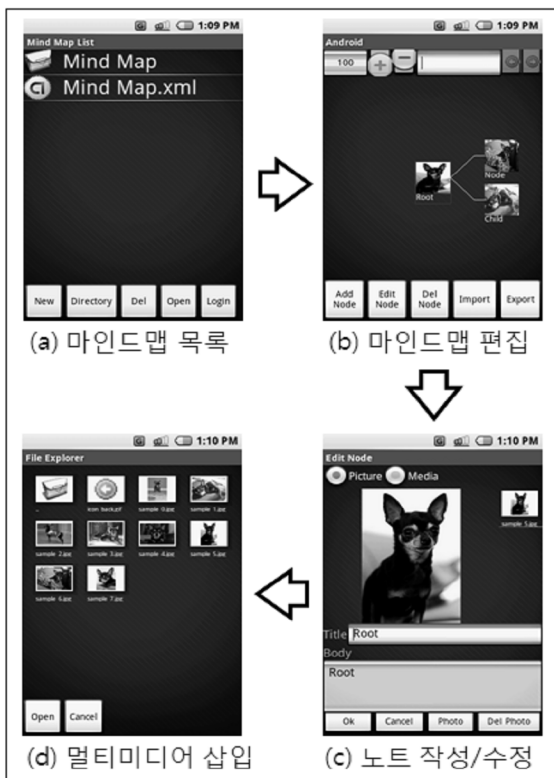
(그림 6) 멀티미디어 삽입 UI 디자인



(그림 7) 멀티미디어 삽입 UI 디자인의 구조관계

실제 실행한 결과는 그림 8의 (d)와 같으며, View와 ViewGroup의 자식 클래스인 LinearLayout, GridView, Button을 원소 이름으로 하고, 위치, 크기, 내용 등을 속성값으로 하여 상세한 모양을 디자인한다.

마인드맵의 실행 과정은 그림 8과 같이 마인드맵을 새로 만들거나 불러와서 노드들에 문자정보를 입력하고 노드의 관계를 명시한 후 그것과 관련된 멀티미디어자료를 연결시키는 순서로 진행된다.



(그림 8) 마인드맵 실행 과정

이중에서 마인드맵의 핵심적인 기능은 그림 8의 (b)의 마인드맵 편집이라 할 수 있는데, 하단의 Add, Edit, Del 버튼을 사용해서 노드를 추가, 수정, 삭제할 수 있다. 또한, Import, Export 기능으로 생성된 마인드맵 노트 자료를 그림 5의 XML파일로 저장하고 불러올 수 있다.

자료구조 형태를 화면에 표시하기 위해서 안드로이드 내에서 제공하는 Canvas 클래스를 사용하였으며, 이 클래스는 그래픽의 확대, 이동, Bitmap 출력, 기본 도형 그리기 등의 기능을 메소드 형태로 제공한다. 화면이 그려질 때는 항상 View.onDraw(Canvas) 메소드가 내부적으로 호출되며, 이를 이용하여 마인드맵을 그려주는 기능과 화면 상단 레이아웃의 '+', '-' 버튼으로 마인드맵의 전체 크기를 확대/축소하는 기능을 구현했고, 상단 텍스트 창을 이용해 마인드맵의 이름 혹은 노드의 내용으로 문자 검색을 할 수 있도록 했다. 또 화면을 드래그 하여 노드를 선택하고 다른 노드로 하위 노드로 이동이 가능한데, 이것은 실제 핸드폰에서는 터치스크린 기능을 이용하여 드래그하기 때문에 View.onTouchEvent(MotionEvent)를 오버라이딩하여 터치스크린을 통한 동작을 구현하였다. 추가, 삭제 등의 각 버튼은 이벤트 핸들러 역할을 하는 View.OnClickListener 클래스의 onClick 메소드를 오버라이딩하여 구현하였다.

5. 결론

본 연구에서는 구글의 개방형 모바일 플랫폼인 안드로이드에서 마인드맵 노트 프로그램을 설계하고 구현하였다. 단순히 문자정보만 저장할 수 있었던 기존의 휴대폰 메모장에 비해 여러 가지 형태의 정보를 저장함은 물론 저장된 메모들을 효과적으로 관리할 수 있도록 하였다. 이를 통해 휴대폰 사용자에게 강력하고도 편리한 기능의 메모 기능을 제공할 수 있을 것으로 기대한다. 추후 연구로 XML로 저장된 마인드맵 노트파일을 WEB에 연동시키거나, 휴대폰 사용자끼리 마인드맵을 공유할 수 있도록 하는 서비스를 개발하는 연구가 필요하다.

[참고문헌]

- [1] http://en.wikipedia.org/wiki/Mind_map
- [2] <http://code.google.com/android>
- [3] <http://code.google.com/android/what-is-android.html>
- [4] <http://code.google.com/android/intro/anatomy.html>
- [5] <http://code.google.com/android/devel/implementing-ui.html>