

논문 2009-46CI-2-8

안드로이드 기반 모바일 정보공유시스템

(Mobile Information Sharing System Based-on Android Mobile Platform)

배 성 호*, 김 우 생**

(Sungho Bae and Woosaeng Kim)

요 약

기존의 휴대폰에 탑재된 메모기능은 단순히 문자정보만을 기록할 수 있고, 공유할 수 없는 일회성 데이터에 불과했다. 이를 보완하기 위해 본 연구에서는 마인드맵(Mindmap) 표기법을 도입하여 멀티미디어 자료를 포함한 메모의 연계성과 관리효율성을 높이고, 백업서버와 웹서버의 연계로 정보를 공유할 수 있는 시스템을 설계하였다. 그리고 이러한 기능들을 구글의 개방형 모바일 플랫폼인 안드로이드(Android)와 리눅스(Linux) 서버의 연계로 구현하였다.

Abstract

The existing note on mobile can store only text data and cannot share the data, which means that the notes stored on mobile are just seasonal or temporary memo. Therefore, this research designs an improved note on mobile and gives a chance for sharing by importing a concept of Mindmap and backup server through the internet. The mobile application is developed based on Android Platform and the server applications are developed based on Linux. These can communicate each other throughout the internet to upload and download some mindmap data.

Keywords : Android, Mobile Information Sharing System, Mindmap.

I. 서 론

현대사회에서 휴대폰은 생활필수품으로 자리 잡고 있다. 최근 휴대폰에는 본래 목적인 통화기능을 넘어서 음악 감상, TV 및 동영상 시청, 게임, 다이어리, 사전, 디지털카메라, 웹브라우저 등의 부가적인 기능들이 탑재되고 있으며, 기능의 종류로 보자면 거의 PC와 비슷한 수준이 되어가고 있다. 이 중에 가장 기본적인 기능인 메모장은 종이나 펜 없이도 꼭 기억해야 할 사항이나, 즉흥적으로 떠오르는 아이디어를 휴대폰에 저장할

수 있게 해준다. 하지만 휴대폰이 멀티미디어를 다룰 수 있음에도 불구하고 메모장에서는 단순히 텍스트 정보만을 나열할 수 있으며, 정보의 백업이나 공유기능은 거의 지원되지 않고 있어 정보의 생산자라기보다는 소비자에 가까운 형태를 가졌다.

본 연구에서는 이런 기존 휴대폰 메모기능을 한층 업그레이드하여 모바일 정보공유시스템을 만들고자 한다. 관리나 공유측면에서 메모기능의 단점을 보완하기 위해 이미지와 언어의 연상 작용을 활용한 노트법인 마인드맵(Mindmap)을 도입하고, TCP기반의 네트워크 백업 및 공유시스템을 구현했다. 이를 구현할 수 있는 모바일 플랫폼으로는 위피(WIFI), 심비안(Symbian), 브루(BREW), 아이폰(iphone), 안드로이드(Android) 등이 있으나, 본 연구에서는 위피 의무화 규정 폐지로 국내 시장에 진출하게 될 다양한 플랫폼 중 가장 최신의 기술이며, 오픈소스를 지향하고 있어 추후 저작권 관련한 비용 면이나 성능 면에서 우수하다고 판단되는 안드로이드

* 정회원, 한국과학기술원 정보통신공학과
(Department of Information & Communication Engineering, Korea Advanced Institute of Science and Technology)

** 정회원, 광운대학교 컴퓨터소프트웨어학과
(Kwangwoon University)

※ 본 논문은 2008년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

접수일자: 2009년2월20일, 수정완료일: 2009년3월6일

이드를 사용했다. 그 외 백업서버와 공유서비스는 각각 리눅스 기반의 데몬(Daemon)과 아파치(Apache)기반의 웹서비스로 개발했다.

본 논문은 다음과 같이 구성되어 있다. II장에서 마인드맵과 안드로이드에 대해 살펴보고, III장에서는 전체적인 모바일공유시스템의 설계와 마인드맵 노트를 저장하고 읽어오기 위한 XML구조, 모바일 마인드맵 노트와 백업서버와의 통신을 위한 네트워크를 설계하고, IV장에서는 3장의 시스템 설계를 바탕으로 모바일 마인드맵 노트와 모바일공유서비스를 구현한다. 마지막 V장에서 결론을 맺고, 향후 연구방향에 대해 논한다.

II. 관련 연구

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^[1]. (그림 1)은 마인드맵 노트의 예로, 노트의 핵심주제를 두고 주변으로 그와 관련된 주제나 아이디어를 가지처럼 뻗치는 형태로 메모하는 방법을 보여준다.

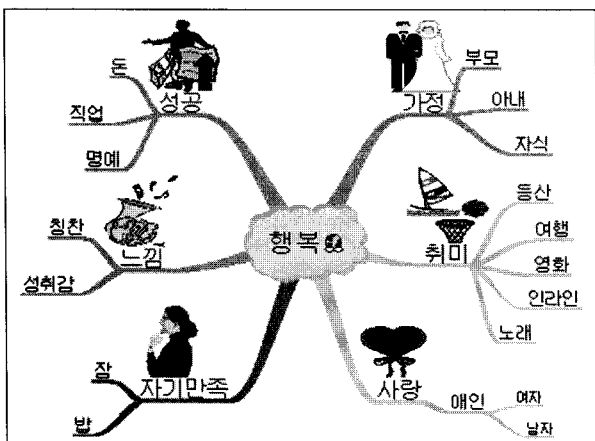


그림 1. 마인드맵 노트의 예
Fig. 1. Example of Mindmap.

2. 안드로이드(Android)

안드로이드는 Google과 OHA(Open Handset Alliance)가 개발한 휴대폰을 위한 소프트웨어 개발 플랫폼

으로, 리눅스를 기반으로 하며, Java를 이용한 응용 프로그램 개발을 지원한다^[2~3]. 안드로이드는 아직 국내에는 출시되지 않았지만 해외에서는 실제 구글폰(G-Phone)에 탑재되어 판매되고 있으며, 최근 위피(WIFI) 의무탑재규정의 폐지로 국내에도 올해 안에 출시될 수 있을 것으로 예상된다.

가. 응용프로그램 구성요소

안드로이드는 Activity, Intent Receiver, Service, Content Provider 이렇게 4가지의 주요한 클래스로 구성되어 있다^[4~5]. 안드로이드 응용프로그램 구성요소가 생성한 이름을 갖고 있지만 그 기능은 (표 1)과 같이 윈도우 같은 GUI(Graphic User Interface) 프로그램과 비슷하다.

Activity는 응용프로그램에서 하나의 화면을 지칭하며, 사용자에게 View와 Event 응답으로 이루어진 인터페이스를 제공한다. 이는 윈도우 응용프로그램의 Form과 같은 역할이며 Form과 다른 점은 History Stack에 있다. 휴대폰은 화면이 크지 않기 때문에 윈도우처럼 여러 개의 창을 동시에 보여주기 힘들다. 그래서 모든 Activity는 화면 전체를 사용하되 (그림 2)처럼 History Stack에 쌓여 있어 현재 작업하고 있는 Activity만이 화면에 보이게 된다. 이 때 History Stack은 각 응용프로그램마다 하나씩 존재하므로, 모든 응용프로그램의 실행상태는 보존된다.

Intent Receiver는 일종의 이벤트 핸들러로서 외부에서 전달받은 Intent를 처리하는 역할을 한다. Intent는 다른 Activity로 이동할 때 필요한 데이터를 담은 클래스이며, 모바일에서는 하나의 Activity에서 이벤트를 처리하기보다는 이벤트가 있을 때 대부분은 새로운 Activity로 전달해야 하기 때문에 포함하는 범위는 윈도우의 이벤트 핸들러와 비슷하다고 할 수 있다.

표 1. Android와 Windows 응용프로그램 구성요소 비교

Table 1. Comparison of Android with Windows in application component.

Android	Windows
Activity	Form
Intent Receiver	Event Handler
Service	Service (Linux Daemon)
Content Provider	ODBC

Service는 윈도우의 서비스나 리눅스의 데몬(Daemon)과 유사하게 백그라운드로 작업을 수행할 필요가 있을 때 사용한다. 기본적으로 안드로이드는 여러 응용프로그램의 상태를 저장하고 있지만, 실제로는 하나의 프로그램만이 동작하게 된다. 즉, 문자를 쓰는 도중 전화를 받았다면 음성통화 프로그램을 실행하기 위해 문자전송 프로그램의 상태를 저장하고 일시정지하게 된다. 하지만 음악을 들으면서 문자를 쓰고 싶은 경우 음악재생 프로그램이 일시정지 된다면 문자를 작성하는 도중에는 음악을 들을 수 없게 된다. 이 때 음악재생 프로그램을 Service로 만들면 백그라운드로 실행되어 두 개의 작업을 동시에 할 수 있게 된다.

Content Provider는 안드로이드에 내장된 SQLite DB를 이용하여 구현되어 있으며, 휴대폰 내의 모든 응용프로그램이 콘텐츠(Content)나 리소스(Resource)를 쉽게 저장하고 공유할 수 있도록 만들어주는 역할을 한다. 이는 ODBC와 비슷한 개념으로 모든 응용프로그램의 데이터관리 인터페이스를 통합해준다. 예를 들어 주소록에 등록된 주소와 구글맵을 이용하여 내비게이션과 같은 길 찾기를 구현하려 한다면 일반적으로 주소록데이터와 구글맵의 위치데이터가 물리적으로 어디에 저장됐는지 알아야 한다. 하지만 안드로이드에서는 다른 응용프로그램의 데이터도 Content Provider를 이용하여 자신의 데이터처럼 쉽게 가져올 수 있도록 지원하기 때문에 그다지 어렵지 않게 구현할 수 있다.

안드로이드의 모든 응용프로그램은 하나 이상의 구성요소 조합으로 이뤄지며, 4가지 구성요소를 모두 포함할 필요는 없다. (그림 2)는 4가지 구성요소를 모두 포함한 주소록 전송프로그램의 예로 Activity로 사용자와 통신하기 위한 UI를 만들고, 주소록을 전송하고자

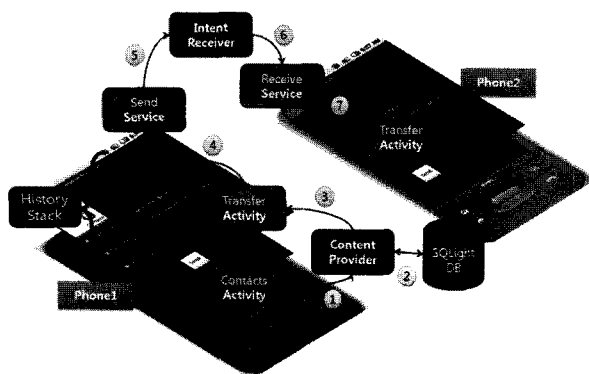


그림 2. 주소록 전송과정에 사용된 구성요소
Fig. 2. Application component used in transmission of contacts.

할 때 ①②③순서로, Content Provider로 정보를 얻어 ④⑤⑥⑦과정처럼 Service와 Intent Receiver를 통해 다른 핸드폰까지 전달할 수 있다.

나. 화면 구성요소

안드로이드의 사용자인터페이스(User Interface)는 View와 ViewGroup 클래스로 구성되며, 이는 (그림 3)과 같이 Composite 패턴^[6]을 이용하여 트리구조를 갖추고 있다. View 클래스는 화면상에서 직사각형 영역의 레이아웃과 내용을 저장하는 자료구조로 화면에서 정렬, 그리기, 포커스 이동, 키의 동작 제어를 담당하는 메서드를 제공한다. ViewGroup은 자신에게 포함된 View와 ViewGroup을 제어하는 기능을 갖고 있으며, 이를 통하면 어떠한 복잡한 UI도 만들 수 있다^[7]. 안드로이드에서는 모든 UI를 JAVA뿐 아니라 XML로도 디자인할 수 있도록 하여 UI디자인 효율성을 높이고 있다.

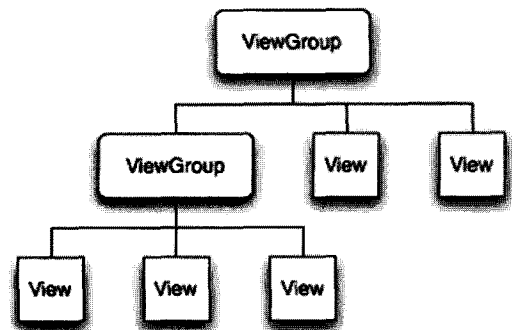


그림 3. Composite 패턴의 화면 구성요소
Fig. 3. User Interface using Composite Pattern.

III. 모바일 정보공유시스템 설계

1. 모바일 정보공유시스템

멀티미디어 기능을 탑재하고 있는 휴대폰이 더 이상 정보시장의 소비자가 아닌 생산자의 역할을 할 수 있도록 하기 위해 (그림 4)와 같은 모바일 정보공유시스템을 설계하였다. 기존의 텍스트 정보만을 저장하던 것을 마인드맵을 이용하여 멀티미디어 정보를 취급할 수 있도록 하고, 휴대폰 손실 및 분실을 대비하여 TCP/IP를 통해 데이터를 백업하고 복구할 수 있는 시스템을 설계했다. 생산된 정보를 일상에서 활용할 수 있도록 하기 위해 마인드맵을 웹서비스로 공유가능하게 하고, 기존 PC에서 동작하는 마인드맵 프로그램인 FreeMind^[8]과 호환가능하게 설계하였다.

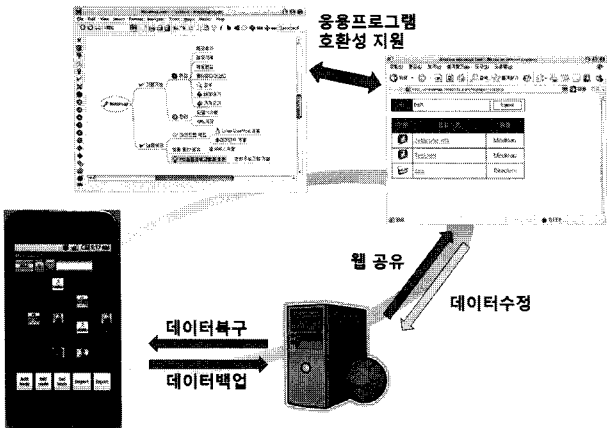


그림 4. 모바일 정보공유시스템 구상도
Fig. 4. Design of Mobile Information Sharing System.

2. 모바일 마인드맵 노트기능

모바일환경에서 마인드맵 노트를 구현하기 위해 (그림 5)와 같은 기능을 설계하였다. 노드(Node)를 하나의 마인드맵 노트의 내부에 삽입되는 아이디어로 정의하고, 노드의 추가, 삭제, 수정, 이동 기능을 설계하였다. 하나의 노드에는 기본적으로 텍스트정보와 다수의 이미지, 동영상 등의 멀티미디어 파일을 삽입할 수 있도록 하고 멀티미디어 데이터는 휴대폰에 저장되어 있는 파일의 주소만을 링크하도록 설계하였다.

또한, 휴대폰의 액정이 주로 세로로 길고 크기가 작음을 고려하여 마인드맵 형태는 오른쪽으로만 퍼져나가는 형태의 트리로 한정하였고, 터치패드를 이용하여 화면의 비율을 확대하거나 축소하는 기능, 화면을 상하좌우로 이동할 수 있는 기능을 설계하였다. 그뿐 아니라 마인드맵 노트의 효율적인 관리를 위해 자식 노드를 분

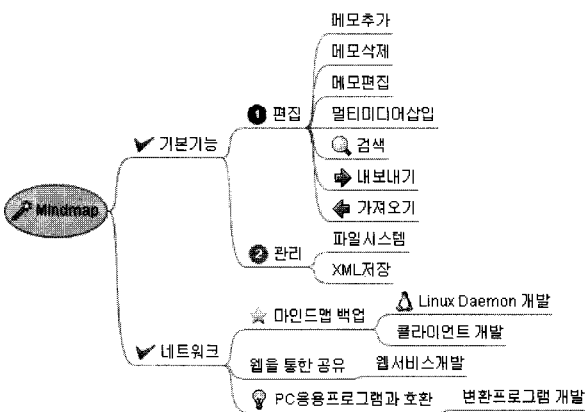


그림 5. 마인드맵 노트의 기능
Fig. 5. Function of Mobile Mindmap Note.

리해서 새로운 마인드맵 노트를 만드는 기능과 마인드맵 노트를 서로 병합하는 기능도 설계하였다.

3. 마인드맵 파일구조

모바일 마인드맵 노트는 인터넷을 통해 다른 사람과 공유할 수 있도록 하는 것이 목적이기 때문에 다른 마인드맵 프로그램과의 호환성을 가지기 위해 XML 파일로 저장하도록 했다.

마인드맵은 하나의 최상위 노드 아래에 다수의 하위 노드를 가지고, 모든 노드는 각각 제목과 본문, 다수의 이미지, 동영상 등의 멀티미디어 자료를 가질 수 있다. 마인드맵이 저장되는 XML 구조는 (그림 6)과 같으며 map 원소는 마인드맵을 시작하는 최상위 원소, node 원소는 노드 하나를 의미한다. 노드의 제목은 title 속성값으로 저장되며, 노드에 저장되는 이미지와 그 외 멀티미디어 파일은 각각 image, media 원소로 저장된다.

```
<!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 6. 마인드맵 XML 파일의 DTD
Fig. 6. DTD format for Mindmap XML.

4. 네트워크 프로토콜

모바일 마인드맵 노트와 백업서버와의 통신을 위해 네트워크 프로토콜을 설계하였다. 일단 무선 기반의 모바일 통신임을 고려했을 때 전송속도보다는 신뢰성확보가 중요하다고 판단되어 TCP를 사용하였고, 마인드맵 및 파일목록 전송프로토콜은 네트워크 및 메모리 자원을 효율적으로 사용하기 위해 가변길이 프로토콜로 설계하였다. (그림 7)은 그 중 하나인 마인드맵 전송프로토콜이며, Header와 Data로 나눠서 설계하였다. 마인드

Header					Data	
	FileSize	Offset	SessionID	MapTitle	Path	Data
1	4	8	8	32	256	1024
						max:128kb

그림 7. 마인드맵 전송프로토콜
Fig. 7. Protocol for transmission of Mindmap data.

맵의 경우 여러 개의 파일로 구성되어 있기에 각 파일마다 그에 대한 응답을 보내도록 설계하였다.

5. 공유서비스 구성요소

마인드맵이 휴대폰에 갠한 정보가 아닌 실제 활용되도록 하기 위해 공유서비스를 설계하였다. 공유서비스는 백업서버의 데이터를 기반으로 하고 있는 일종의 웹 서비스로 크게 3가지 기능을 중심으로 설계하였다. 그 기능은 백업된 마인드맵 데이터를 웹으로 볼 수 있는 기능, PC로 압축된 마인드맵을 다운받는 기능, 기존 PC용 마인드맵 프로그램 파일로 변환기능이며, 이를 통해 모바일에서 생산된 데이터도 정보로써 활용될 수 있도록 설계하였다.

IV. 모바일 정보공유시스템 구현

1. 모바일 마인드맵 노트

모바일 마인드맵 노트는 MS Windows XP에 Android SDK m5-r15를 설치한 후, Eclipse와 ADT 0.4.0을 사용하여 Java로 개발하였다.

안드로이드 UI는 기존의 다른 모바일 플랫폼과는 달리 모든 UI 관련 클래스를 XML의 각 원소에 매핑시켜 XML로 디자인이 가능하게 함으로써 복잡한 디자인도

좀 더 직관적으로 세부적인 조정이 가능하다. (그림 8)과 같이 View와 ViewGroup을 상속받은 LinearLayout, GridView, Button을 원소 이름으로 하고 각 위치, 크기, 내용 속성을 정의하면 실행시 클래스 자동매핑과정을 거쳐 (그림 9-d)와 같은 화면을 볼 수 있다.

모바일 마인드맵 노트의 일반적인 실행과정은 (그림 9)와 같으며, 새로운 마인드맵을 생성하거나 기존 마인드맵을 불러와서 해당 노트에 문자정보를 입력하고 멀티미디어 정보를 삽입한 후 노트간의 관계를 명시하게 된다. 이중에서 가장 핵심적인 기능은 (그림 9-b)의 마인드맵 편집기능이라 할 수 있다. 모든 사용자가 보이는 화면은 하나의 Activity에 View를 덧씌운 것이며, (그림 9-b)의 마인드맵 편집기능의 경우 화면 이동, 확대/축소, 마인드맵 그리기 등의 동적이고 특수한 기능이 필요했기에 안드로이드 내에서 제공하는 Canvas 클래스를 이용하여 구현하였다. 이는 그래픽의 확대, 이동, 비트맵 출력, 도형그리기 등의 기능을 갖고 있으며, 화면을 그리는 시점에 내부적으로 항상 View.onDraw(Canvas)가 호출하므로 이를 통해 필요한 동적인 기능을 구현할 수 있었다. 동적인 기능 중에서 화면을 상하좌우로 이동하는 것을 구현하기 위해서는 터치스크린을 활용할 필요가 있어 View.onTouchEvent(MotionEvent)를 오버라이딩

```
<?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 8. 멀티미디어 삽입 UI 디자인

Fig. 8. UI Design of Multimedia List Activity using XML.

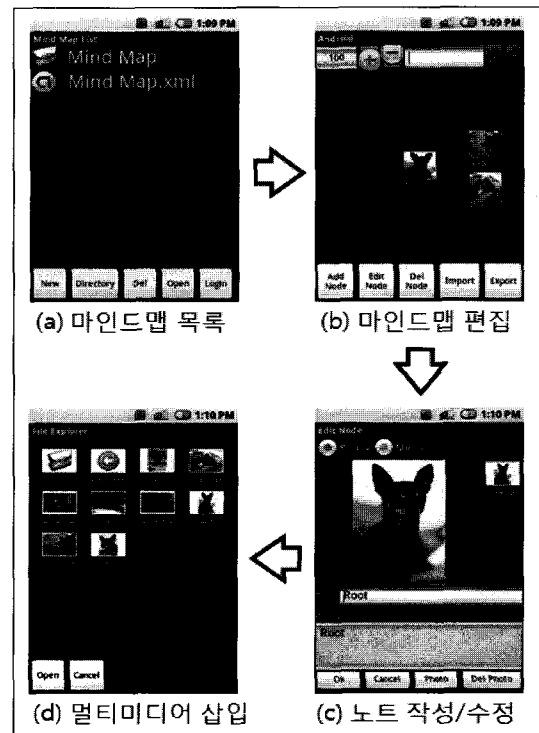


그림 9. 모바일 마인드맵 노트 실행과정

Fig. 9. Running process of Mobile Mindmap Note.

하기도 했다. (그림 9-c)에서 Ok버튼을 클릭시 내부적으로 Title, Body, Photo, Media를 포함하는 Intent를 만들고 다음 Activity에서 Intent Receiver가 이 정보를 해석하여 Content Provider를 통해 저장하는 과정을 거쳐 저장하도록 구현하였으며, 버튼 클릭을 통한 Activity 이동의 경우 대부분 같은 방법으로 구현할 수 있다. 또한, 네트워크 백업기능의 경우 백업서버와 TCP 통신을 통해 구현했으며 세션파일이 저장되어 있을 경우 자동으로 로그인되도록 하는 세션 로그인도 구현하였다.

2. 마인드맵 서버

마인드맵 서버는 Fedora 6(Linux)과 mysql 5.0.27을 설치한 후, C언어를 이용하여 데몬(Daemon) 프로세스 형태로 제작하였다. 리눅스 내에서 가장 먼저 생성되는 프로세스가 init(PID=1)이고, 이 프로세스가 모든 데몬 프로세스의 부모가 되며, 일반적인 응용프로그램의 프로세스는 데몬 프로세스의 복제로 생성되는 프로세스이다. 즉, 백그라운드로 실행되는 데몬 프로세스가 되기 위해서는 (그림 10)과 같은 과정을 거쳐 init 프로세스의 직속 자식 프로세스로 존재해야 하고, 터미널을 가지지 않는 자신만의 세션을 가져야 한다.

데몬 프로세스는 터미널을 가지지 않고, 백그라운드로 실행되기 때문에 실제 모바일 마인드맵 노트와 통신하는 과정을 테스트하기 위해서는 전송되는 정보를 기록할 필요가 있다. 그래서 이런 통신내역을 로그파일로 저장하기 위해 로그 파일만을 담당하는 함수를 printf() 함수와 유사한 형태로 구현했다. 전송과정 이외에 로그인 정보나 전송된 마인드맵 내역들은 mysql을 이용하여 관리했고, 실제 전송된 마인드맵 및 멀티미디어 데이터는 파일로 저장하도록 구현했다.

```
int main() {
    pid_t pid;
    if (( pid = fork()) < 0) exit(0);
    else if(pid != 0) exit(0); // 부모프로세스 종료
    signal(SIGHUP, SIG_IGN); // 터미널종료 signal 무시
    close(0); close(1); close(2);
    setsid(); // 새로운 세션
    while(1) {
        // 프로그램 본체
    }
}
```

그림 10. 데몬 프로세스 제작 틀
Fig. 10. Format for Daemon process.

3. 공유서비스

공유서비스는 단순히 백업을 위해 저장된 마인드맵 데이터를 PC를 통해 공유할 수 있는 통로를 만들어주기 위한 것으로, 별다른 설치 없이 가장 쉽게 접근할 수 있는 웹서비스로 구현하였다. 마인드맵 서버와는 독립적으로 존재할 수 있지만 구현의 편의를 위해 마인드맵 서버와 물리적으로 같은 곳에 구현하였다. Fedora 6(Linux), mysql 5.0.27, apache 2.2.6, php 5.1.6을 설치한 후, php와 python을 이용하여 구현하였다.

공유서비스의 전체적인 틀은 php와 mysql을 연동하여 구현하였고, 변환기능이나 다운로드 기능은 php에서 shell_exec()함수를 이용하여 외부로 구현된 python 프로그램을 호출하도록 만들었다. 그리고 공유서비스의 UI 중 (그림 11)와 같이 마인드맵이 보이는 부분은 Javascript의 XML DOMParser^[9]와 DHTML(Dynamic HTML)^[10]을 이용하여 동적으로 생성되도록 구현하였다.

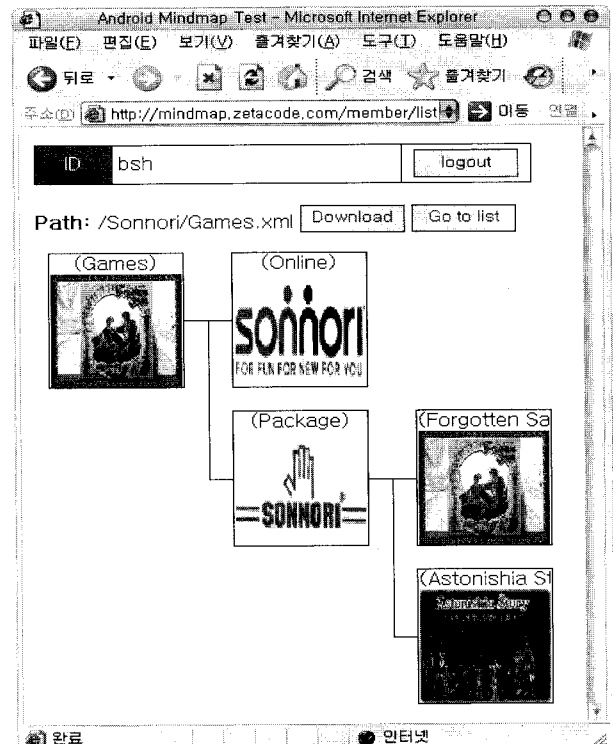


그림 11. 마인드맵 공유서비스
Fig. 11. Sharing service of Mindmap Note.

V. 결 론

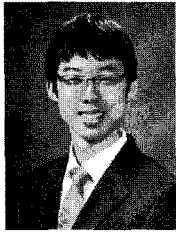
본 연구에서는 구글의 개방형 모바일 플랫폼인 안드로이드에서 모바일 마인드맵 노트를 설계하고, 이를 통

해 생산되는 정보를 효율적으로 공유하기 위한 모바일 정보공유시스템을 설계 및 구현하였다. 이로써 단순히 문자만을 저장할 수 있었던 메모의 개념을 바꾸고 메모를 하나의 통합된 정보로써 저장하고 관리할 수 있게 하였다. 또한, 공유시스템을 설계함으로써 휴대폰이 정보를 소비하는 것만이 아닌 생산주체로 역할을 할 수 있는 기반을 만들었다. 이는 사용자에게 유용하고 편리한 메모기능을 제공함은 물론 휴대폰을 통한 정보공유를 보편화시킬 수 있을 것으로 기대되며, 개발자측면에서는 오픈 모바일플랫폼인 안드로이드를 이용함으로써 로열티 등의 추가비용을 절감시키고, 개발의 편의성을 향상시킬 수 있을 것으로 기대된다. 이 연구를 활성화하기 위해서는 추가로 휴대폰 단말간의 통신을 통한 공유나 최신 버전으로 소스 업그레이드하는 기술, 안드로이드 플랫폼을 실제 보드에 포팅 하는 기술 등에 대한 연구가 필요하다.

참 고 문 헌

- [1] http://en.wikipedia.org/wiki/Mind_map
- [2] <http://developer.android.com/>
- [3] <http://developer.android.com/guide/basics/what-is-android.html>
- [4] <http://developer.android.com/guide/topics/fundamentals.html>
- [5] <http://developer.android.com/guide/topics/ui/declaring-layout.html>
- [6] http://en.wikipedia.org/wiki/Composite_pattern
- [7] <http://developer.android.com/guide/topics/ui/index.html>
- [8] <http://freemind.sourceforge.net/>
- [9] [http://msdn.microsoft.com/en-us/library/ms766487\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms766487(VS.85).aspx)
- [10] <http://en.wikipedia.org/wiki/DHTML>

저 자 소 개



배 성 호(정회원)
2009년 광운대학교
컴퓨터소프트웨어학과
학사 졸업.
2009년~현재 한국과학기술원
(KAIST) 정보통신공학과
석사 재학.

<주관심분야 : 운영체제, USN>



김 우 생(정회원)
1985년 서울대학교 수료
U. of Texas at Austin
전산학과 학사 졸업.
1987년 U. of Minnesota
전산학과 석사 졸업.
1991년 U. of Minnesota 전산학과
박사 졸업.

1992년~현재 광운대학교 컴퓨터소프트웨어학과
교수.

<주관심분야 : 멀티미디어, 데이터베이스>