

# 오픈 API를 이용한 마인드맵 프로그램의 설계 및 구현

## Design and Implementation of Mind map program using Open API

이선웅\*, 이해림\*\*, 김유두\*\*, 문일영\*

Seon-Ung Lee\*, Hye-Rim Lee\*\*, Yoo-Doo Kim\*\* and Il-Young Moon\*

### 요 약

본 논문에서는 매쉬업 기능을 제공하기 위하여 오픈 API를 활용한 마인드맵 프로그램에 대해서 제안하였다. 웹의 패러다임이 웹 2.0으로 넘어가면서 오픈 API를 이용한 매쉬업이 크게 활용되고 있다. 매쉬업이 웹에서의 서비스만으로 한정하지 않고 본다면 새로운 아이디어나 정보를 산출하는 방법 중 하나로 좋은 방법이다. 이러한 방법을 체계적으로 만들어 둔 것이 바로 마인드맵이다. 본 논문에서는 오픈 API를 이용하여 모바일 기기에서 활용할 수 있는 마인드맵 응용 프로그램을 설계 및 구현하였다. 이는 현재 입력된 단어와 연관된 검색어를 표시하고 웹 브라우저와 연동되어 모바일 기기의 효율성을 높일 수 있다. 따라서 연구의 중요성은 크다고 할 수 있으며, 빠른 상용화를 통해서 시장 창출의 효과를 가지고 올 수 있으리라 예상됩니다.

### Abstract

In this paper, it is proposed a mind map program using open API to provide mashup function. Web paradigm is changing to Web 2.0. So mashup using open API is much applied. Mashup is good method for not only web service, but making new ideas or informations. It is mind map that was made systematical like this method. In this paper, a mind map application based on mobile that provides mashup function implemented for modern people that mostly process their business during movement.

Key words : Mind map, Mashup, Open API

### I. 서 론

새로운 정보를 만드는 방법 중 가장 많이 쓰이는 방법 하나는 인터넷의 무한한 정보를 서로 합치는 것이다. 특히 매쉬업(Mashup)이 웹 2.0의 구성 요소로 주목을 받고 있는데, 이는 웹으로 제공하고 있는 정보와 서비스를 융합하여 새로운 소프트웨어나 서비스, 데이터베이스 등을 만드는 것을 말한다.

마인드맵(Mind map)이란 중심 키워드를 중심으로 단어, 아이디어, 작업 등의 항목을 연결하여 원형으로 정렬한 그림이다[1]. 이 그림은 새로운 아이디어를 산출하거나, 조직, 분류, 시각화하는데, 주로 교육, 조직, 문제 해결, 의사 결정에서 많이 사용된다. 이렇게 아이디어를 산출하는 것은 현재 정보를 만들어내는 작업과 유사하다.

Google이나 Yahoo, Microsoft 등이 제공하는 지도

\* 한국기술교육대학교 인터넷미디어공학부(School of Internet Media Eng., Korea University of Technology and Education)

\*\* 한국기술교육대학교 대학원 정보미디어공학과(Department of Information Media Eng., Korea University of Technology and Education)

· 제1저자 (First Author) : 이선웅

· 투고일자 : 2009년 2월 3일

· 심사(수정)일자 : 2009년 2월 4일 (수정일자 : 2009년 2월 16일)

· 게재일자 : 2009년 2월 28일

서비스, 인터넷 서점 Amazon이 제공하는 상품 정보 등, 자사의 기술을 웹 서비스로서 API(Application Programming Interface)를 공개하는 경우가 늘고 있으며, 이들 기능에 독자적인 사용자 인터페이스를 융합하여 새로운 서비스를 제공하고 있다[2],[3]. 또한 문명의 발달과 함께 사람이 하루에 처리해야 하는 작업의 양이 늘어가는 것과 동시에 이동하는 시간도 늘어났다. 오히려 이동하는 시간이 더 많이 늘어남으로 인하여 이동하는 도중에 일을 처리해야 하는 상황이 생겨나고, 결과적으로 그러한 요구에 맞춘 휴대폰, PDA(Personal Digital Assistants), 네비게이션, 노트북 등과 같은 모바일 장비들이 등장하였다. 이러한 장비들의 등장을 통해 현대인들은 이동 중에도 전화와 문서작업, 스케줄관리, 정보 검색 등을 통해 많은 일들을 하고 있다. 이러한 모바일 장비에서 정보를 만들어내는 일은 점점 많아질 것이고 매쉬업의 기능을 제공하는 응용 프로그램의 필요도 증가할 것이다.

본 논문에서는 매쉬업 기능을 제공하기 위하여 오픈 API를 활용한 모바일 기반의 마인드맵 응용 프로그램에 대해서 제안하고자 한다.

제 II장에서는 마인드맵 구성 요소들에 대한 설명을 한다. 제 III장에서는 응용프로그램에 필요한 자료구조를, 제 IV장에서는 모바일 기기에 적합한 출력 알고리즘을, 제 V장에서는 오픈 API를 이용하여 연관 검색어링크를 통한 매쉬업 기능을 살펴본다. 마지막으로 제 VI장에서는 결론과 향후 발전 과제를 제시하도록 한다.

## II. 마인드맵 구성요소

마인드맵의 구조는 크게 두 가지 요소로 구분할 수 있으며, 아이디어 또는 개념을 나타내는 컨셉(Concept)과 컨셉들을 이어주는 가지(Branch)로 설명할 수 있다[1]. 컴퓨터 자료구조에서 그래프 형태의 구조이다. 그리고 인터넷은 실제로 하이퍼링크를 통해서 연관된 페이지가 연결되어 있다. 이 구조를 보면 하나의 정보를 갖는 페이지와 이어주는 링크 그리고 관련된 정보를 갖는 페이지가 서로 이어진 그래프 구조를 갖고 있다. 따라서 현재 정보의 바다라고 불

리는 인터넷의 구조와 마인드맵의 구조는 정보를 만들고 저장하기 쉬운 그래프 자료구조가 사용됨을 알 수 있다.

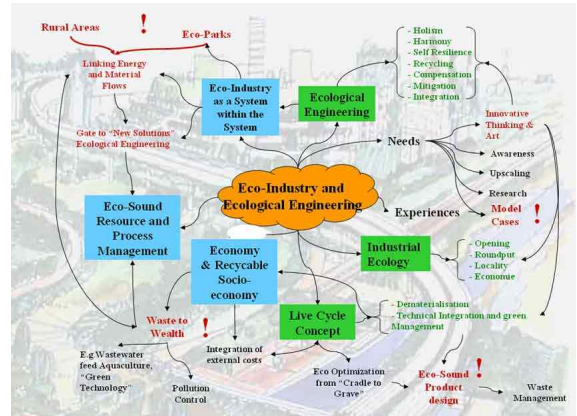


그림 1. 마인드맵의 예  
Fig. 1. Example of mind map

컨셉이란 아이디어 또는 개념을 나타내는 간략한 문장 또는 단어로 나타내어진다. 본 논문에서는 모바일 장치의 디스플레이가 작다는 점을 고려하여 단어만을 표현하고 단어에 대한 설명은 숨겨두는 것을 기본 전제로 한다. 가지는 마인드맵에서 각 컨셉들을 연결하는 다리역할 및 생각의 흐름을 나타내는 역할을 한다. 가지 자체로 큰 의미를 갖지는 않지만 각 컨셉의 연결을 통해 발상의 전환 또는 문제를 해결하는 아이디어를 유도하기 때문에 마인드맵에서의 해결책을 제시하는 중요한 요소라고 할 수 있다.

## III. 자료 구조

본 논문에서 전반적인 구현 환경은 Windows Mobile CE 기반의 MFC(Microsoft Foundation Class)를 전제로 진행한다[4].

### 3-1 컨셉

컨셉은 그림 2와 같이 하나의 중심 컨셉을 두고 파생된다. 중심 컨셉은 파생 컨셉들에 대한 정보를 갖고 있어야 한다. 또, 파생 컨셉에서 다시 파생되어진 컨셉을 가질 수 있다. 즉, 파생된 컨셉이 중심 컨셉이

되어 자신에게서 파생된 개념들을 가질 수 있다. 자료 구조적으로 분석해 보면, 각 개념을 인식하는 식별자와 의미를 갖는 개념 단어가 저장되어야 한다. 위에서 잠시 언급되었던 개념에 대한 설명을 숨기기 때문에 별도로 저장하여 보여줄 공간과 개념의 모양을 저장하는 공간이 필요하다.

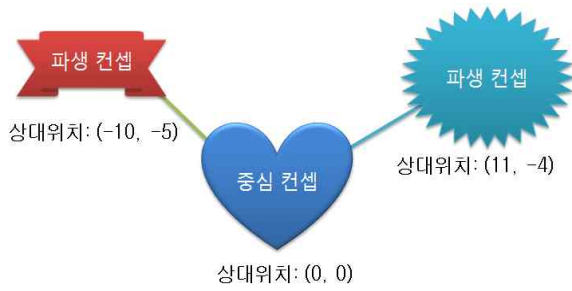


그림 2. 개념 개념도

Fig. 2. Concept map of concepts

본 논문에서는 각 파생 개념들의 위치를 중심 개념이 갖고 있도록 하여 출력할 때 위치를 상대적으로 계산하도록 구현하였다. 이 파생 개념들의 식별자와 위치 정보를 저장하는 리스트가 필요하다.

현재까지 언급된 중요한 개념 정보들만 포함하여 구조를 만들어 보면 그림 3과 같이 표현되며, 개념 클래스의 인터페이스를 설명하면 표 1과 같다.

● Data Structures – Concept Class

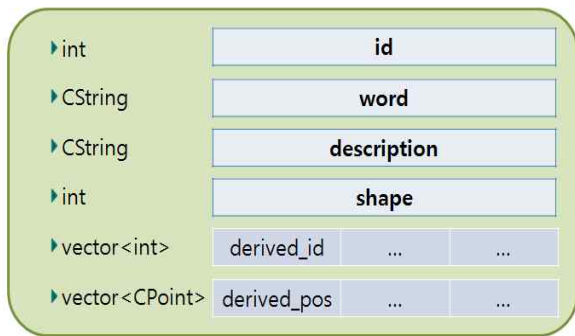


그림 3. 개념 클래스 구조

Fig. 3. Structure of the concept class

표 1의 함수들은 개념 클래스에 대한 공개된 함수들이고, 이 함수들을 통해 개념들을 제어할 수 있게 된다. 식별자의 경우 가져오는 인터페이스만 있는 이유는 생성자에서만 식별자를 입력할 수 있도록 하기 위함이다. 파생 개념의 경우 STL(Standard Template

Library)인 vector를 사용하는데, 이를 이용한 내부 구현은 배열과 비슷하다. 이를 제외한 대부분 함수는 직접 대입하거나 값을 읽어오는 간단한 작업을 하고 있다.

3-2 개념 관리자

개념의 자료 구조만으로는 마인드맵을 효과적으로 관리하거나 보여줄 수 없으므로 개념을 관리하는 개체가 필요하다. 본문에서는 이를 개념 관리자라 하여 설명하도록 한다.

개념 관리자는 개념 클래스의 인터페이스를 이용하여 개념들을 제어하고, 식별자 분배 관리, 개념들의 리스트 관리, 중심 개념 그리고 출력 시 위치 계산을 줄이기 위한 중심 개념에 대한 개념들의 상대 위치 리스트를 관리하게 된다.

표 1. 개념 클래스 인터페이스  
Table 1. The concept class interface

생성자	Concept(int ID, CString concept, int figure);
개념 식별자	GetConceptID() const;
개념 단어	SetConceptWord(CString concept); GetConceptWord() const;
개념 설명	SetDescription(CString descript); GetDescription() const;
파생 개념	AddDerivedConcept(int id, CPoint relPos); DelDerivedConcept(int id);
모양	SetFigure(int figure); GetFigure();

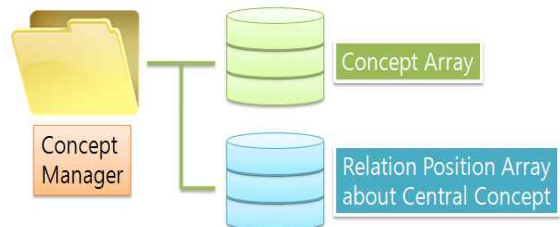


그림 4. 개념 관리자 개념도

Fig. 4. Concept map of the concept manager

위에서 언급된 정보들을 갖는 컨셉 관리자의 구조를 정의하면 그림 5와 같다. 그림을 보면 컨셉 클래스와 컨셉 관리자 클래스에서 상대위치 리스트는 vector를 사용하고 컨셉 리스트에 대해서는 CArray를 사용하고 있다. MFC를 기반으로 응용 프로그램을 개발하다 보면 겪게 되는 문제가 동적 리스트를 다루는 문제에서 발생하게 된다. 모든 클래스가 CObject를 상속받게 되어있는데, CObject의 생성자는 외부에서 접근이 불가능하기 때문에 CArray와 같은 컬렉션 클래스에서 저장될 데이터 타입에서 또 다른 MFC 클래스를 멤버 변수로 사용하고 있다면, 대입 연산자(operator =)를 각 MFC 멤버 변수를 대입할 수 있게 public으로 overriding해야 한다.

● Data Structures – Concept Manager Class

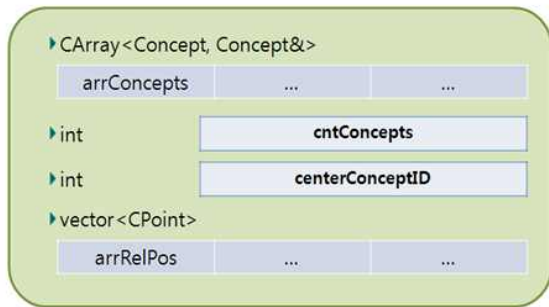


그림 5. 컨셉 관리자 클래스 구조

Fig. 5. Structure of the concept manager class

표 2. 컨셉 관리자 클래스 인터페이스

Table 2. The concept manager class interface

<b>컨셉 관리</b>
AddConcept(CString word, CPoint pos, int figure);
DelConcept(int ID);
ChangeCenterConcept(int ID);
MoveConceptPos(int ID, CPoint dPos);
SetConceptFigure(int ID, int figure);
<b>이벤트/출력 관련</b>
GetClosestID(const CRect& rect);
GetClickID(CPoint point);
GetConceptsInRect(CRect rect, int** arr);
<b>직렬화</b>
Serialize(CArchive& ar);
<b>초기화</b>
Init();

위의 표 2는 컨셉 관리자의 인터페이스를 설명한

내용이다. 컨셉 관리자 클래스에서는 해당 아이디를 갖는 컨셉에 접근을 용이하게 하기 위해서 리스트가 추가될 때, 식별자의 오름차순으로 정렬한다. 컨셉이 추가될 때의 알고리즘을 간단히 pseudo code로 작성하면 표 3과 같다.

3-3 어플리케이션

어플리케이션에서의 기본적인 구조는 그림 6과 같다. 모든 컨셉에 대한 권한을 컨셉 관리자가 캡슐화하여 가지고 있고, 문서가 컨셉에 대한 이벤트를 받을 때마다 컨셉 관리자를 통하여 제어한다.

표 3. AddConcept pseudo code

Table 3. AddConcept pseudo code

```

AddConcept(CString word, CPoint pos, int figure)
{
    if (컨셉 리스트의 개수 == 0) {
        Concept concept(1, word, figure);
        컨셉 리스트 추가(concept);
        상대 위치 리스트 추가(CPoint(0,0));
        중심 컨셉 식별자 변경(1);
    }
    else if (컨셉 리스트 개수 != 리스트의 마지막 컨셉 식별자) {
        할당되지 않은 가장 작은 식별자와 삽입위치 찾음;
        Concept concept(찾은 식별자, word, figure);
        가장 가까운 컨셉과 파생 관계 연결 및 갱신;
        컨셉 리스트 삽입(삽입위치, concept);
        상대 위치 리스트 삽입(삽입위치, pos);
    }
    else {
        Concept concept(리스트 개수+1, word, figure);
        가장 가까운 컨셉과 파생 관계 연결 및 갱신;
        컨셉 리스트 추가(concept);
        상대 위치 리스트 추가(pos);
    }
}
    
```

뷰는 화면에 변화가 발생할 때마다 DrawThread에 비트맵 갱신 메시지를 보내고 DrawThead는 컨셉 관리자에게서 화면에 출력될 만큼의 데이터만 가져와서 그리게 되며, 기본적으로 파일 입출력은 MFC에서

쉽게 할 수 있는 직렬화를 이용한다. 직렬화에서는 CObject를 상속받은 MFC 클래스 이어야 하며 다음 매크로를 선언해줘야 한다.

```
// 클래스 선언부 (*.h)
DECLARE_SERIAL(ConceptManager)
// 클래스 정의부 (*.cpp)
IMPLEMENT_SERIAL(ConceptManager,CObject,1)
```

직렬화를 선언한 다음 Serialize 함수를 overriding하게 되면 표 4와 같이 문서에서 간단히 직렬화를 통한 파일 저장 및 불러오기를 할 수 있다.

```
CArchive ar( &f, CArchive::load);
cm.Serialize(ar);
ar.Close();
f.Close();
```

IV. 출력 알고리즘

본 장에서는 마인드맵을 출력하는 알고리즘에 대하여 설명한다. 먼저 모바일 기기 상에서 모든 좌표를 상대위치로 변환하여 저장하고, 화면의 중점과 중심 컨셉과의 거리를 이용하여 화면에 출력한다. 그리고 화면에 출력될 컨셉들을 계산하여 해당 컨셉들만 출력하도록 한다.

4-1 상대위치(Relational Position)

그림 7에서 2차원 좌표계를 확인할 수 있으며, 그림 7과 같은 상황에서 A에 대한 B의 상대 위치는 (5, -3)이다. 역으로 B에 대한 A의 상대 위치는 (-5, 3)이다. 이렇게 상대 위치는 자신을 원점으로 계산하여 대상의 위치를 설명할 수 있다.

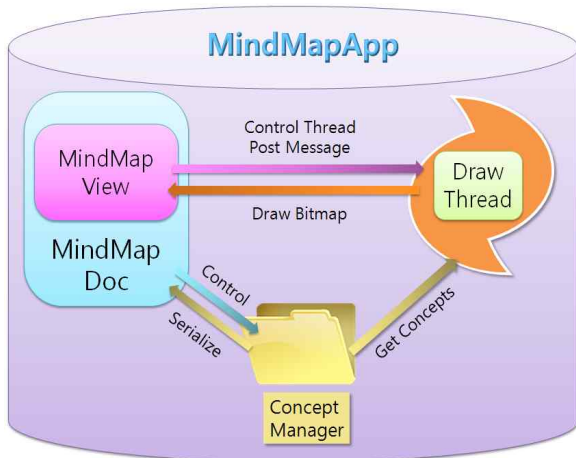


그림 6. 마인드맵 어플리케이션 개념도  
Fig. 6. Concept map of the mind map application

표 4. 직렬화 예제 코드

Table 4. Example code of serialization

```
ConceptManager cm();
CFile f;
if (!f.Open( "C:\\test.dat", CFile::modeCreate | CFile::modeWrite)) {
    return;
}

// 저장
CArchive ar( &f, CArchive::store);
cm.Serialize(ar);
ar.Close();
f.Close();

// 불러오기
```

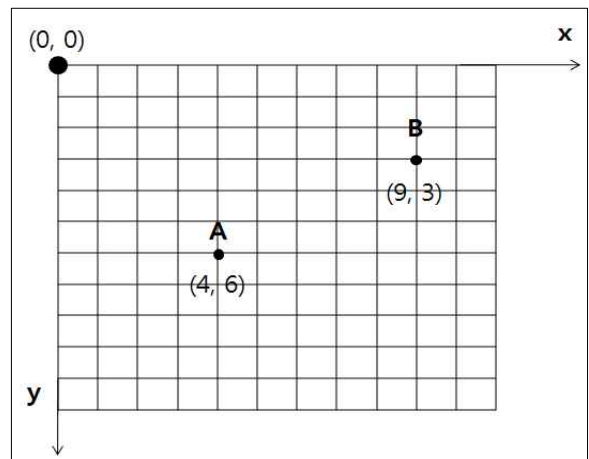


그림 7. 2차원 좌표계  
Fig. 7. Coordinate system in second-dimension

마인드맵을 구현할 때 상대 위치를 사용하는 것은 중심 컨셉을 바꿀 수 있기 때문이다. 모바일 장치에서의 출력은 화면의 제약 때문에 많은 내용을 한 번에 출력할 수가 없다. 따라서 깊게 마인드맵이 그려진다면 처음의 중심 컨셉은 보이지 않고 현재 출력된



단어들의 상위 컨셉 1~2단계 정도까지만 나타내어진다. 이에 따라 자유롭게 중심 컨셉을 바꾸어 다양한 방법의 작성이 가능하도록 구현하였다.

상대위치를 사용하는 또 다른 이유로는 마인드맵을 그리는 공간의 자유도를 높이기 때문이다. 기본적으로 그림을 그리기 위해서 비트맵을 사용하는데, 여기서는 한정된 공간(미리 정해진 크기만큼)만 그릴 수밖에 없다. 따라서 유형으로 일정 비트맵을 정해놓고 그 공간 안에서만 그리는 방식이 아니라 거의 무한한 공간에 그릴 수 있고, 화면 범위 안의 컨셉만 골라서 출력할 수 있기 때문이다. 본문에서는 View와 컨셉 관리자, DrawThread에서 모두 상대위치로 변환하여 좌표계의 일관성을 유지하였다. 화면 범위 안의 컨셉만 골라내는 일은 DrawThread에서 컨셉 관리자의 GetConceptsInRect 함수를 통해 컨셉 식별자 배열을 받아 화면에 출력될 비트맵을 그리게 된다.

#### 4-2 Viewpoint 기법

Viewpoint 기법이란 모바일 장치의 화면을 하나의 카메라처럼 생각할 수 있게 하는 기법이며, 마인드맵의 부분을 이동하여 볼 수도 있고 Zoom in/out하여 볼 수 있게 한다. 화면을 이동하는 방법은 화면에서 드래그를 통하여 원하는 위치로 이동한다. 위치에서 컨셉에 대한 이벤트를 처리할 때도 일관성을 유지하기 위해 중심 컨셉에 대한 현재 화면 중점의 상대 위치를 갖고 있어야 한다. Zoom in/out 기능을 구현하기 위해서 다양한 기술이 사용될 수 있으나, 비교적 간단하고 쉽게 사용되는 비트맵 스트레칭을 통해서 구현하였다. 팝업 슬라이더를 통해서 원하는 크기에 맞추면 DrawThread에 확대 값을 보내는 메시지를 동적으로 보내어 화면에 적용될 수 있게 하였다. 확대되거나 축소된 화면은 비트맵을 늘린 것이므로 원래의 이미지와는 조금 다르게 나올 수 있다. 그림 8과 그림 9는 실제로 구현된 확대/축소된 화면을 보여준다. DrawThread에서는 그림 8과 그림 9와 같이 확대/축소 비율과 viewpoint 중점의 상대위치를 계산하여 화면의 영역은 CRect 자료형으로 컨셉 관리자의 GetConceptInRect 함수로 넘긴다. GetConceptInRect 함수는 식별자순으로 정렬된 리스트를 순서대로 찾아 보면서 상대위치로 지정된 사각영역 안의 컨셉들을

찾아내고 배열을 작성하여 이를 반환한다.

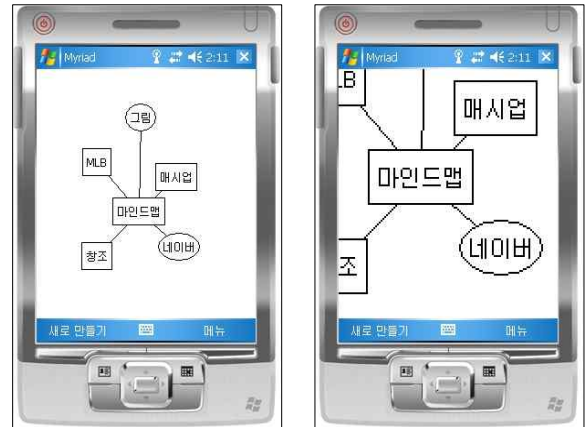


그림 8. 확대된 화면  
Fig. 8. Enlarged screen

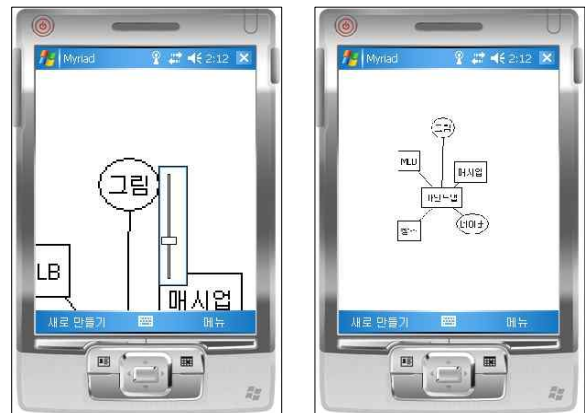


그림 9. 축소된 화면  
Fig. 9. Reduced screen

## V. 매쉬업 (Mashup)

### 5-1 오픈 API

오픈 API는 SOAP(Simple Object Access Protocol), Javascript 등과 같은 웹 기술을 이용하여 웹사이트끼리 서로 상호작용할 수 있게 하는 기술들의 집합체라고 할 수 있다[5]. 웹사이트에서 제공되는 기술을 이용하여 다른 서비스와 연동시킬 수 있다[6].

현재 Google에서는 Google Maps, Google AJAX (Asynchronous JavaScript and XML) Search, Google Earth 등의 서비스를 오픈 API로 제공하고 있고, 이를 응용한 여러 웹 응용 프로그램들이 개발되어 서비스

되고 있다. 점점 오픈 API를 서비스하는 곳은 증가하고 있다. 특히 Picasa와 flickr, Youtube, Yahoo와 Google 등의 서비스를 매쉬업하여 이미지와 플래시 게임, 동영상 등을 서비스하는 Cooliris는 3D로 체험할 수 있게 UI(User Interface)를 구현하였고, 오픈 API를 잘 사용한 표본으로 나타나고 있다. Firefox, Internet Explorer, Safari 브라우저와 각 OS의 버전에 맞게 배포되고 있어 쉽게 접할 수 있다.

본 논문에서는 네이버의 오픈 API를 사용하여 구현하였다. Google에 비해 많이 공개되어 있지 않고 개발도 많이 되어있지 않지만, 우리나라 사용자들에게 적합한 연관 검색어 결과를 보여줄 것이라 예상된다.

5-2 연관 검색어 링크

연관 검색어는 실제로 사용자들로 하여금 흥미를 끌거나 관심이 있어 입력된 검색어 다음으로 많이 찾은 검색어가 랭크된다. 이러한 서비스는 사용자가 생각지 못했던 정보를 얻을 수 있게 해준다.

본 논문에서는 오픈 API 중 연관 검색어를 이용하여 새로운 컨셉이 입력되었을 때, 그 컨셉 단어로 요청한 응답을 처리하는 것을 구현한다. 네이버에서는 연관 검색어를 추천 검색어라는 이름으로 서비스하고 있다. 네이버 오픈 API 사이트에서 일반 키를 받아야 하고, 네이버 오픈 API 사이트에서 제공되는 키는 일반 키와 지도 키 두 가지가 있다. 지도를 연동할 것이 아니므로 일반 키를 받으며, 일반 키를 받은 후에는 다음과 같은 네이버 추천 검색어 샘플 URL에 접근가능하다.

<http://openapi.naver.com/search?key=test&query=nhn&target=recmd>

위의 key=test 부분을 앞서 받은 기본 키를 입력하여야 결과 화면을 볼 수 있다. 결과는 XML(eXtensible Markup Language) 형식으로 전송되는데 개발언어에서 알맞게 파싱하여 원하는 값을 얻어야 한다. 그림 10은 실제 프로그램에서 새로운 컨셉을 입력 하였을 때 추천 검색어가 출력되는 화면이다. 그리고 표 5는 상위 3개의 연관 검색어를 가져오는 알고리즘을 나

타낸다.

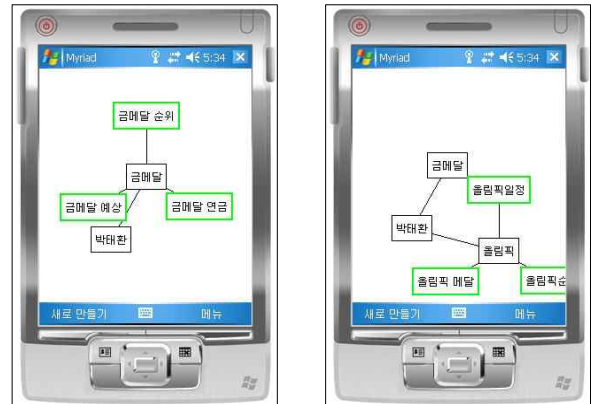


그림 10. 네이버 추천 검색어 연동 결과

Fig. 10. Interlocked screen with related keyword of 'Naver'

표 5. 상위 3개의 연관 검색어를 얻는 알고리즘  
Table 5. Algorithm to get top-ranked 3 related keyword

```

GetLinkItems(word) {
    sock.connect("openapi.naver.com", 80);
    word_utf = toUTF(word);
    param="GET /search?key=" + key + "&query=" +
        word_utf + "&target=recmd HTTP/1.0\r\n\r\n";
    sock.send(param, param.getLength());
    sock.receive(buf, bufsize);
    keywords = parsexml(buf);
    for (i=0; i<3; I++) top3[i] = keywords[i];
    return top3[i];
}
    
```

연관 검색어의 결과는 최대 10개까지 받아올 수 있지만, 화면의 제약으로 인하여 추천 검색어는 3개를 최대로 출력하게 하였다. 추천 검색어는 웹 브라우저와 연동되어 클릭하면 해당 검색 페이지가 열리며, 이로 인해 사용자는 추천 검색어를 통한 정보를 얻고 매쉬업하여 자신만의 정보를 만들 수 있다.

VI. 결 론

본 논문에서는 오픈 API를 이용하여 모바일 기기에서 활용할 수 있는 마인드맵 응용 프로그램을 설계

및 구현하였다. 매쉬업 기능이 추가된 마인드맵 프로그램은 연관 검색어를 통해 얻은 정보들을 매쉬업하여 새로운 정보를 창출할 수 있게 한다. 따라서 이동 중 많은 업무를 처리하는 현대인들에게 많은 도움이 될 것이라 판단된다[7]. 특히 연관 검색어를 표시하고 웹 브라우저와 연동되는 부분은 모바일 기기의 제한된 UI에서도 효율성을 높일 수 있어 핵심이 되는 부분이다. 정보가 인터넷을 통해 공유되면서 정보를 창출하기 위하여 더 고수준의 창의력이 필요로 하게 되는 데, 모바일 기반의 마인드맵은 이러한 문제들을 해결함에 있어서 매력적인 도구가 될 것이라 예상된다. 또한, 마인드맵이 사용자 중심의 프로그램인 만큼 보편적인 UI가 추가되고, 사용자 환경 개선을 위한 UX(User eXperience)가 적용된 모바일 응용 프로그램이 좀 더 나은 서비스를 지원할 것으로 기대되며, 무엇보다도 모바일 단말의 작은 화면에서는 2차원으로 보는 것보다 3D를 활용한 화면구성이 정보의 생산적 측면에서 앞으로 해결해 나가야할 문제로 예상된다.

### 참 고 문 헌

- [1] [http://en.wikipedia.org/wiki/Mind\\_map](http://en.wikipedia.org/wiki/Mind_map)
- [2] <http://ko.wikipedia.org/wiki/%EB%A7%A4%EC%89%AC%EC%97%85>
- [3] PC PLUS, Google Maps mashup: Link geocoding data with Google Maps, Vol.-No.256, 2007
- [4] <http://msdn.microsoft.com/>
- [5] [http://en.wikipedia.org/wiki/Open\\_API](http://en.wikipedia.org/wiki/Open_API)
- [6] Duane Merrill, Mashups: The new breed of Web app, IBM DeveloperWorks, Oct, 2006
- [7] Holt Adams and John Gerken, Choosing between mashups and traditional Web applications, IBM DeveloperWorks, Jul, 2008

### 이 선 웅 (李善雄)



2004년 3월~현재 : 한국기술교육대학교  
인터넷미디어공학부 재학  
관심분야 : 모바일 플랫폼, Open API,  
Mashup, UI

### 이 혜 림 (李憶琳)



2008년 8월 : 한국기술교육대학교 인터넷  
미디어공학부 졸업 (공학사)  
2008년 9월 ~ 현재 : 한국기술교육대학교 대학원  
정보미디어공학과 재학 (공학석사)  
관심분야 : 무선 TCP, 무선 메쉬 네트  
워크, 라우팅 프로토콜

### 김 유 두 (金裕斗)



2007년 2월 : 한국기술교육대학교 인터넷  
공학 졸업 (공학사)  
2009년 2월 : 한국기술교육대학교 대학원  
정보미디어공학과 졸업 (공학석사)  
2009년 3월~현재 : 한국기술교육대학교 대학원  
정보미디어공학과 (박사과정)

관심분야 : 무선 네트워크, 라우팅 프로토콜, 모바일 콘텐츠

### 문 일 영 (文日永)



2000년 2월 : 한국항공대학교  
항공통신정보공학과 (공학사)  
2002년 2월 : 한국항공대학교 대학원  
항공통신정보공학과 (공학석사)  
2005년 2월 : 한국항공대학교 대학원  
정보통신공학과 졸업(공학박사)  
2004년 ~2005년 : 한국정보문화진흥

원 선임연구원

2005년 3월~현재 : 한국기술교육대학교 인터넷미디어공학부  
조교수

관심분야:무선 인터넷 응용, 무선 인터넷, 모바일 IP