

논문 2011-2-7

OpenAPI를 이용하여 간결함과 즉시성을 가진 검색 매쉬업

Mashup for Instant and Simple Search using OpenAPIs

황기태*

Kitae Hwang

요 약 본 논문은 사용자가 작업하는 문서에서 검색하고자 하는 검색어를 빠르고 단순하게, 그리고 다양한 검색의 범위를 조절할 수 있는 검색 방법을 제안하고 구현한 사례를 소개한다. 기존 웹 사이트 검색 방식은 하던 작업을 멈추고 웹 브라우저를 실행하고 사이트에 접속하며 키워드를 입력하는 지루한 과정이 필요하다. 또한 검색의 결과도 장황하며, 사전, 이미지, 동영상 등 동시 검색이 필요한 경우 여러 사이트에서 여러 번 검색하는 불편함이 있다. 본 논문은 사용자가 작업하는 어떤 문서에서도 간단히 검색어를 선택하고 CTRL키와 마우스 왼쪽 버튼을 동시에 누르는 것으로 검색이 이루어지도록 한다. 이를 위해 웹 서비스를 이용할 수 있는 Open API를 이용하여 여러 검색 웹 사이트의 검색 기능을 동시에 통합적으로 활용하는 매쉬업 검색 소프트웨어를 구현한다. 본 논문에서 개발한 검색 소프트웨어는 검색의 단순 즉시성, 결과의 간결성, 검색 범위의 다양성, 기능의 확장성뿐만 아니라 작업하는 어떤 문서에도 검색이 가능한 매쉬업의 좋은 사례로 평가된다.

Abstract This paper introduces a new search model which can retrieve information instantly and easily about a given search word on Internet and control search spaces diversely. In the existing web searches, because the user has to stop his work, run web browsers, connect to special internet sites, and type the search word by keyboard. it is a long way and not a simple way. Also the search result is too much. The user suffers difficulties of multiple searches for different categories such as dictionary, image, video clips, etc. The software implemented in this paper enables for the user to select search word simply at any documents on the screen and make search started by simply pressing CTRL key and left button of the mouse simultaneously. The software is a mash up style using the Open API which is provided by a web site and can be freely available in any other sites. By calling different Open APIs of several web sites, the mash-up software can integrate search functions of many different sites. We evaluate that the software implemented in this paper will be a good reference of mash-up as one with characteristics of easy and instant search, simple result, diversity of search category, and scalability of search functions.

Key Words : Search, Open API, Mashup

I. 서 론

인터넷 검색을 시행하고자 하면 사람들은 구글^[1], 야

후^[2], 네이버^[3], 다음^[4] 등 포털 사이트에 접속하고 사이트에서 제공하는 사전(dictionary) 검색이나 이미지 검색을 수행한다. 혹은 사진이나 이미지만을 전문으로 다루는 플리커^[5] 등의 사이트에 접속하여 전문적인 콘텐츠를 검색한다.

이러한 검색 형태는 처음부터 집중적으로 어떤 내용

*중신회원, 한성대학교 컴퓨터공학과
접수일자: 2010.11.26, 수정일자: 2011.2.15
게재확정일자: 2011.4.15

을 검색하고자 목적으로 시작한 경우이다. 그러나 사용자가 문서 작업 중 비교적 빠르고 쉽게 간단한 검색 정보를 얻고자 하는 경우 기존의 웹 검색 포털 사이트를 방문하는 검색 방식은 부적합하다. 예를 들어 사용자가 컴퓨터에서 어떤 문서를 읽거나 작성하거나, 혹은 웹 서핑을 하고 있다가 갑자기 모르는 단어를 발견하여 그 뜻을 알고 싶다고 하자. 그러면 하던 작업을 멈추고 웹 브라우저를 실행한 후 네이버 등과 같은 포털 사이트에 접속하고 검색 키워드를 입력하거나 사전 검색 페이지를 열어 검색어를 입력하는 길고도 지루한 과정을 거쳐야 한다면 여간 불편하지 않다. 또한 검색 결과도 필요 이상으로 과도하다. 사용자는 검색을 그냥 포기하는 것을 택할지 모른다.

본 논문은 사용자가 문서 작업 중 쉽고 편리하게 이용할 수 있는 검색 응용프로그램을 제안하고 구현한 내용을 소개한다. 키 아이디어는 즉시성, 간결성, 맞춤형에 있다. 즉시성은 컴퓨터에서 임의의 작업을 하는 도중 복잡한 과정을 거치지 않고 한 번의 키 입력만으로 원하는 단어를 바로 검색할 수 있음을 말하며, 간결성은 검색 결과가 간결성을 뜻하고, 맞춤형은 사용자가 원하는 종류의 검색 범위는 스스로 간단히 선택할 수 있음을 뜻한다.

본 논문에서 제안하는 검색 응용프로그램을 MISO (Mashup for Instant Search using Open APIs)라고 명명하며 개념은 그림 1과 같다. 이 응용프로그램(MISO)은 백그라운드로 시스템 상에서 실행된다. 사용자가 컴퓨터에서 어떤 작업도중 “박지성” 단어를 발견하고 박지성에 관련된 뉴스, 동영상, 사전적 의미, 사진 이미지, 동영상 등을 간단히 찾고 싶은 것이다. 사용자는 발견한 “박지성”이라는 단어를 마우스로 선택하고 간단한 축약키 (CTRL+마우스왼쪽버튼)를 입력하면 바로 검색이 이루어진다. MISO는 각 검색 포털 사이트에서 제공하는 Open API^[6]를 동시에 호출하고 이들이 전송해오는 결과를 높은 랭킹 순으로 일부분만 사용자에게 동시에 보여준다.

본 검색 응용은 매쉬업^[7] 형태의 응용이다. 사전, 이미지, 뉴스 등 검색의 범위는 사용자가 메뉴에서 간단히 설정하며 새로운 분야에 대한 검색이 필요한 경우 해당 사이트의 OpenAPI를 호출하는 부분만을 추가하면 되기 때문에 본 검색 응용프로그램은 확장성이 높은 장점이 있다.

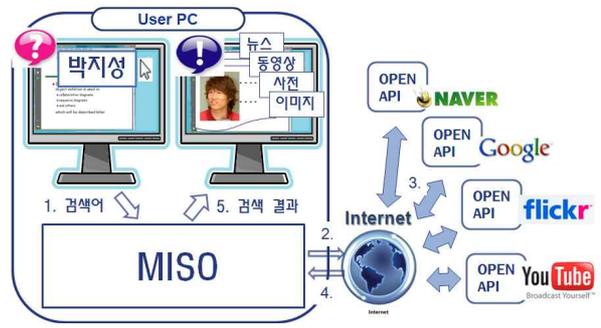


그림 1. 매쉬업 검색 방법의 개념
Fig. 1. Concept of Mashup Search

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 기존 연구와 배경 지식에 대해 논하고, 3장에서는 본 논문에서 제안된 검색 방법에 대한 설계 내용을, 4장에서는 구현과 시연 결과를 보이고 5장에서 결론을 맺는다.

II. 관련 연구

1. 유사 검색 소프트웨어 사례

검색 엔진에 관한 연구를 중심으로 검색의 용이성과 효율성을 위한 많은 연구들이 진행되어 왔다^[8]. 본 논문의 연구 대상은 새로운 검색 기법을 개발하거나 검색 결과의 정확성을 높여서 검색자의 만족도를 높이기 위한 연구가 아니다. 본 논문은 매쉬업 기법으로 기존 검색 사이트를 활용하여 사용자가 원하는 검색 결과를 빠르고 간결하게 얻어낼 수 있는 응용 레벨의 검색 매쉬업의 설계 및 개발에 관한 것이다.



그림 2. 로지텍 SetPoint의 검색 설정
Fig. 2. Search Configuration at Logitech SetPoint

본 논문의 목적과 유사한 과거의 사례를 살펴 보자. Logitech SetPoint^[9] 라는 소프트웨어로서 로지텍 마우스 드라이버와 함께 배포되며, 로지텍 마우스에 달린 검색 버튼을 누르면 선택한 텍스트를 지정한 검색 사이트의 검색 결과를 가져다 마이크로소프트의 인터넷 익스플로러에 출력시켜 주는 기능이다. SetPoint 소프트웨어는 그림 2와 같이 설정한다.

SetPoint의 장점은 마우스에 검색 버튼이 달려 있어 한 번의 클릭만으로 검색을 시행하는 즉시성이 있으나 로지텍 마우스에 제한되는 한계가 있으며, 검색이 이루어지는 포털 사이트를 오직 하나만 지정하므로 검색의 결과가 단편적이라는 단점이 있다.

다른 사례로 야후의 Mini^[10]라는 소프트웨어로서, 인터넷 서핑이나 문서 작업 중 모르는 단어에 빨간 미니펜을 끌어다 놓으면 한영, 영영, 한자, 국어, 영어, 백과 사전 등의 검색 결과를 미니 창에 자동으로 보여주는 서비스를 지원한다. Mini를 사용하는 사례는 그림 3과 같다. Mini는 해당 단어에 대한 사전적 정보만 검색 가능하며, 야후 사이트에서 제공되는 정보만 이용 가능하므로 검색의 범위가 제한적이며 융통성이 부족한 단점이 있다.

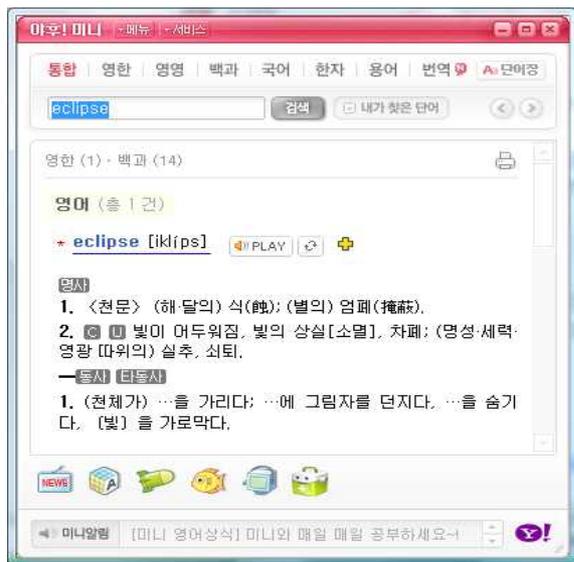


그림 3. 야후 Mini의 작동
Fig. 3. Operation of Yahoo Mini

2. 매쉬업과 Open API

하나 이상의 응용 서비스를 결합하여 새로운 서비스 혹은 응용을 만들어 내는 것을 매쉬업(Mashup)이라고 부른다. 여러 개의 웹 서비스를 이용하여 하나의 새로운

서비스를 만들어 낼 수 있다. 매쉬업의 예를 들면 구글의 지도 서비스와 부동산 정보 서비스를 결합하면 매물로나온 부동산의 위치를 지도 상에 바로 출력하는 응용 서비스를 간단히 만들 수 있다.

이러한 매쉬업이 가능하게 하는 기술적인 요소가 바로 Open API이다. Open API란 기업이 자신이 제공하는 웹 서비스의 일부를 API 형태로 공개함으로써 누구나 웹 서비스를 이용할 수 있도록 하는 것이다. 구글의 지도 서비스, 위성 사진 서비스, 아마존의 도서 정보 서비스, 야후나 네이버의 검색 서비스 등 많은 사이트들이 Open API 방식으로 서비스를 공개하고 있다.

Open API가 가능하도록 하는 통신 기술은 REST(Representational State Transfer). XML/SOAP, XML-RPC, RSS/ATOM^[11] 등이 있다. 최근에는 이중 REST 방식이 많이 사용된다. REST는 Open API 요청 시 HTTP 요청방식을 사용하고 그 응답은 XML 형태로 전송하는 방식이다.

보통의 경우 Open API를 이용하기 위해서는 사전에 사이트에서 허용하는 key 값을 받아야 하며 이 key 값을 Open API 요청문에 넣어 주어야 한다.

III. MISO 설계

1. 설계 원칙

본 논문에서 설계 구현한 MISO의 설계 원칙은 크게 다음과 같다.

① 쉽고 편리한 단어 검색

사용자가 컴퓨터 상에 출력된 임의의 문서 상의 문자열을 선택한 후 CTRL키와 마우스 왼쪽 버튼을 동시에 누르는 한 번의 입력으로 검색이 이루어지도록 한다.

② 사용자 맞춤형 검색

한 번의 검색으로 검색어에 대한 사전 정보, 뉴스 정보, 백과사전 정보, 사진, 동영상, 번역 등의 검색을 동시에 진행할 수 있으며 사용자가 검색의 범위를 간단히 선택할 수 있다.

③ 간결한 검색 결과 제공

검색의 결과는 웹 검색 사이트에서 제공하는 소수의

최고 랭킹 정보만을 출력하여 검색 결과의 복잡도를 줄인다.

④ 윈도우 상의 모든 문서에서 검색 단어 선택 가능
기존의 검색 방법들은 대부분 검색어를 키로 입력하거나, 특정 하드웨어나 특정 소프트웨어를 사용하여야 하는 불편함이 있었다. 본 논문에서 추구하는 검색 방법은 화면에 출력된 어떤 문서에서도 검색어를 선택할 수 있도록 한다. 그러므로 웹 브라우저, PDF, 워드, 한글, 기타 어떤 문서를 읽는 도중이라도 그 문서 내에 있는 어떤 문자열이라도 검색어로 설정하여 검색이 일어나게 한다.

⑤ 작은 크기의 응용

OpenAPI를 이용함으로써 검색 엔진을 탑재하지 않고 매쉬업 형태로 작성하여 소프트웨어의 크기를 최소한으로 줄여 부담 없는 소프트웨어를 작성한다.

⑥ 높은 확장성

새로운 검색 범위가 필요하면 이에 따라 쉽게 검색 범위를 넓힐 수 있는 확장성을 가지게 한다. OpenAPI를 이용하는 특징상 새로운 검색 범위가 요구되면 정보를 가진 사이트에서 제공하는 몇 개의 간단한 OpenAPI를 이용하여 새로운 콘텐츠 검색을 쉽게 포함할 수 있다.

2. 기능 정의

가. 검색 타입

본 연구에서는 다음 6 가지의 검색 타입을 제공한다. MISO는 높은 확장성으로 인해 검색 타입을 추가하기는 매우 쉽다.

- 사전 정보 - 검색어에 대한 사전 정보
- 백과 사전 - 검색어에 대한 백과 사전 정보
- 뉴스 - 검색어에 대한 관련 최신 뉴스
- 사진 - 검색어에 대한 사진 이미지
- 동영상 - 검색어와 관련된 동영상
- 번역 - 검색어의 한글, 영어, 중국어, 일본어의 상호 번역 기능

사용자는 6 개의 검색 타입의 모든 정보를 검색할 수도 있고 선택적으로 오직 한 개의 타입만 검색할 수도 있다.

나. 검색어 직접 입력

사용자는 문서에게 검색어를 선택할 수 있으며 또한 검색어를 응용프로그램에 직접 입력하여 검색을 요청할 수 있다.

3. 소프트웨어 구조 및 동작 설계

MISO의 구성은 그림 4와 같이 Hooking Module과 Search Module의 2 부분으로 구성된다. MISO는 백그라운드에서 실행되며 사용자가 임의의 소프트웨어를 사용하고 있을 때 검색이 일어나는 과정은 다음과 같다.

① 검색 이벤트 발생

사용자가 윈도우에서 한글 등 임의의 소프트웨어 사용 중 마우스로 단어를 선택한 후 시스템에 검색을 요청하기 위해 CTRL 키와 동시에 마우스의 왼쪽 버튼을 누른다. 앞으로 이 키 조합을 검색 이벤트로 부른다. Hook DLL은 윈도우 운영체제에 입력되는 모든 입력 사항을 염탐하는 후킹 작업을 하도록 된 개발된 DLL 이다. Hook DLL은 검색 이벤트의 발생을 알아채게 되면 Hook Server 모듈에게 검색 이벤트의 발생을 알린다.

② 검색 이벤트 전달

Hook DLL은 후킹을 통해 탐지한 검색 이벤트의 발생을 Hook Server에게 알린다.

③ CTRL+C 키 삽입

Hook Server는 운영체제의 시스템 키 버퍼에 CTRL 키와 C 키 값을 삽입하여 마치 사용자가 키보드로 CTRL+C 키가 눌러진 것처럼 보이게 한다.

④ WM_CUT 메시지 보냄

윈도우 운영체제는 키 버퍼에서 CTRL+C 키를 발견하고 현재 사용자가 사용 중인 응용프로그램에게 WM_CUT 메시지를 보낸다. 윈도우 운영체제는 사용자가 CTRL+C 키를 입력한 것으로 오해하고 현재 사용 중인 응용프로그램에게 선택된 문자열을 컷(CUT)하여 시스템 클립보드에 삽입하도록 지시한다.

⑤ 검색어 복사

WM_CUT 메시지를 받은 윈도우 응용프로그램은 선택된 문자열을 시스템 클립보드에 복사한다. 이 과정은

사용자 모르게 응용프로그램 내부적으로 진행된다.

⑥ 검색어 읽기

Hook Server는 과정 ③을 실행하고 일정 시간을 기다린 후 시스템 클립보드에 담겨 있을 검색어를 가지고 온다.

⑦ 검색어 전송

Hook Server 는 검색어를 MISO Search Engine에게 전송한다.

⑧ Open API 호출

MISO Search Engine은 내부에 SThread 라는 스레드를 만들고 이 스레드는 Open API를 이용하여 검색어를 적절한 사이트에 전달하고 검색을 지시한다.

⑨ 검색 결과 전송

사이트는 검색을 진행하고 그 결과를 보낸다.

⑩ 검색 결과 출력

UI 모듈은 검색 결과들을 디스플레이에 출력한다.

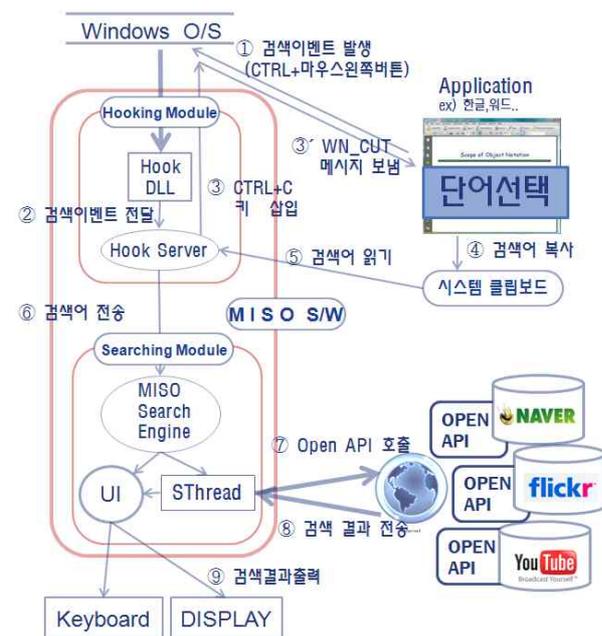


그림 4. MISO 소프트웨어 구조 설계
Fig. 4. Software Architecture of MISO

IV. MISO의 구현

1. 시스템 독립적인 부분과 의존적인 부분으로 분리 구현

MISO는 시스템 의존적인 부분과 시스템 독립적인 부분으로 분리 구현되었다. Hooking Module은 시스템 의존적인 부분으로서 윈도우 운영체제와 밀접하게 연관된다. 이 부분은 후킹과 시스템 클립보드 사용, 시스템 키보드 버퍼에 대한 접근 등을 위해 C++ 언어로 개발하고 실행 모듈인 HookServer.exe 파일과 후킹을 위한 HookDLL.dll 파일로 분리하여 구현하였다.

그러나 Searching Module은 운영체제 독립적인 부분이고 통신이나 GUI 사용이 상대적으로 편리한 자바 언어로 구현한다. 그리고 Hooking Module을 바로 자바 가상머신 상에서 실행이 가능하도록 하나의 실행 파일인 miso.exe로 컴파일하였다.

그러므로 MISO는 표 1과 같이 구성되며 HookServer.exe와 MISO.exe는 소켓을 이용하여 통신한다. MISO의 실행은 HoolServer로부터 시작한다.

표 1. MISO의 구현 모듈

Table 1. Implemented Modules of MISO software

모듈	타입	목적
HookDLL	C++로 작성된 DLL	검색 이벤트인 CTRL+마우스왼쪽버튼 입력을 후킹하는 목적
HookServer	C++로 작성된 EXE	후킹한 검색어를 MISO.exe로 전송한다. UI가 전혀 없다.
MISO	자바로 작성된 EXE	HookServer.exe 모듈로부터 검색어를 전달 받고 Open API를 이용하여 검색을 지시하고 결과를 자바 GUI를 통해 출력한다. 바로 윈도우에서 바로 실행할 수 있는 exe로 컴파일함

2. 실행 사례

구현의 자세한 설명에 이해를 돕기 위해 MISO 소프트웨어 실행 사례를 설명한다.

가. 트레이에 아이콘 형태로 실행

윈도우에서 MISO 소프트웨어를 실행하면 그림 5와 같이 윈도우의 하단 오른쪽의 트레이 속에 아이콘 형태로 실행을 시작한다. 트레이의 아이콘은 MISO.exe에서 출력하며 트레이 상의 소프트웨어로 작동시키는 것도 MISO.exe에서 처리한다.



그림 5. MISO의 트레이에서의 실행
Fig. 5. Execution of MISO in the system tray

나. 검색 제어 창

그림 5의 트레이에서 MISO의 아이콘을 더블클릭하면 그림 6과 같이 검색 타입을 선택하거나 검색어를 직접 입력하여 검색을 시행할 수 있는 검색 제어 창이 출력된다.

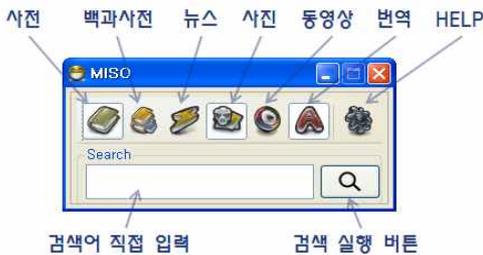


그림 6. MISO의 검색 제어 창
Fig. 6. Control Panel of MISO

총 6 가지의 검색 타입을 선택할 수 있으며 이들은 모두 토글 버튼으로 만들어져 있다. 현재 그림 6의 경우에는 사전, 뉴스, 사진, 번역 정보만 검색하도록 사용자가 선택한 경우이다.

다. 검색 실행

사용자가 PDF 문서를 읽다가 그림 7과 같이 "correlation" 의 뜻을 몰라 마우스로 문자열 영역을 선택하고 CTRL키+마우스왼쪽버튼을 클릭하였다.

그 결과 그림 7과 같이 매우 빠른 속도로 검색이 이루어지고 검색 결과 창이 출력된다. 검색 결과는 사전, 뉴스, 사진, 번역의 4 가지로서 그림 6에서 선택한 타입을 대상으로 검색한 결과이다.

3. Hooking Module 구현

가. 프로그램 시작

HookServer는 하나의 C++ 파일로 작성되었으며 WinMain() 메소드를 가지고 있어 MISO의 실행 진입점으로서의 역할을 수행한다. WinMain()이 시작되면 HookDLL을 호출하여 그림 8에 보이는 MouseHookProc 함수를 전역 훅(global hook)으로 등록하도록 한다. 그리고 나서 Searching Module(MISO.exe)을 자식 프로세스로 실행시킨다.

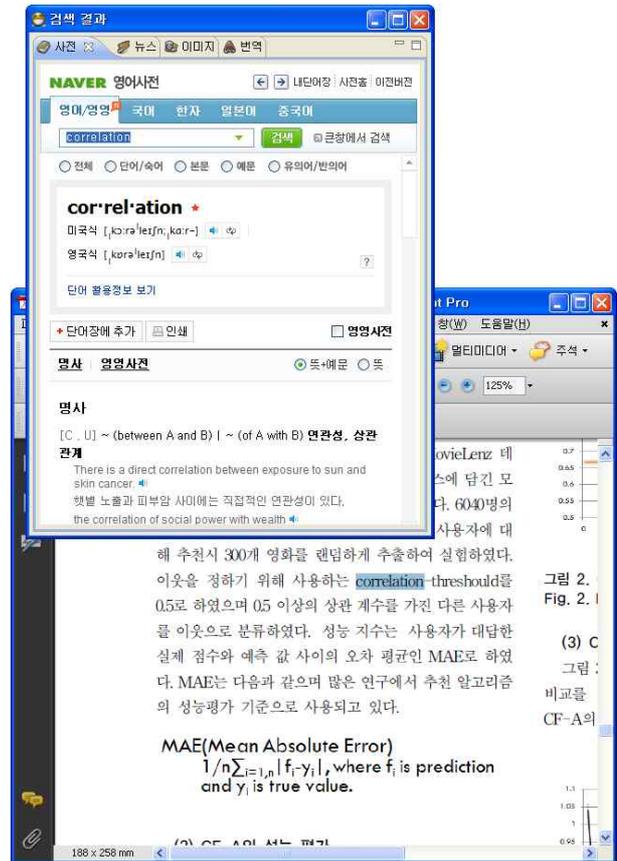


그림 7. "correlation"의 검색 결과
Fig. 7. Result of Search for "correlation"

```
SetWindowsHookEx(WH_MOUSE, MouseHookProc, hModule, NULL);

LRESULT CALLBACK MouseHookProc(int nCode, WPARAM wParam,
                                LPARAM lParam)
{
    if (nCode >= 0) {
        switch (wParam) {
            case WM_LBUTTONDOWN :
                if (GetAsyncKeyState(VK_CONTROL) < 0) {
                    SendMessage(hHookServer, WM_USER + 1,
                                wParam, lParam);
                    return 1;
                }
            case WM_LBUTTONUP :
                if (GetAsyncKeyState(VK_CONTROL) < 0) {
                    return 1;
                }
        }
    }
    return CallNextHookEx(hMouseHook, nCode, wParam, lParam);
}
```

그림 8. 후킹 구현 함수
Fig. 8. Functions implementing hooking

나. 후킹 구현

후킹 구현을 위해 HookDLL.dll을 구현하였다. 일반적으로 DLL을 만드는 방법으로 프로그램을 작성하며 다음

과 같이 간단히 윈도우 운영체제의 메시지 후킹을 위한 함수(MouseHookProc)를 등록하면 된다.

다. 이벤트의 흐름

사용자가 CTRL키 + 마우스 왼쪽버튼을 클릭하여 HookServer가 검색 이벤트의 발생을 알게 되었을 때, 마치 CTRL+C가 눌러진 것처럼 윈도우 운영체제의 키버퍼에 CTRL키와 C키를 입력시킨다.

그러면 윈도우 운영체제는 자연스럽게 현재 실행중인 활성화된 응용프로그램(포커스를 가진 응용프로그램)에게 WM_CUT 메시지를 전달하게 되며 현재 실행 중인 응용프로그램의 WM_CUT 처리 루틴은 현재 선택된 문자열을 클립보드에 복사한다.

라. Hooking Server와 MISO Searching Server 사이의 통신

Hooking Server와 MISO Searching Server는 UDP 소켓을 이용하여 통신하도록 구현하였다. 두 서버 모듈은 각각 독립적인 실행 모듈로 생성되므로 이들 사이의 통신 방법으로는 소켓이 적당하며 동일한 컴퓨터 내에 있기 때문에 패킷을 잃어버릴 가능성이 없으므로 굳이 통신 부하가 상대적으로 높은 TCP를 사용할 필요가 없다.

Hooking Server는 현재 시스템 클립보드에 있는 검색어를 가지고 와서 UDP로 MISO Search Server에게 보낸다. 검색어의 크기는 보통 몇 바이트 수준이다. 만일 사용자가 그림 6에서 번역을 선택한 경우 긴 문장을 선택하여 검색어로 사용할 수도 있다. 그러나 이 경우에도 역시 문자들이므로 검색어의 크기는 매우 작다고 할 수 있다. 그러므로 통신에 따른 오버헤드는 거의 없다고 봐도 무관하다.

4. Searching Module 구현

Searching Module은 자바로 작성하였으며 Eclipse의 RCP(Rich Client Platform)^[12]사용하여 작성하였다. RCP는 크로스 플랫폼 데스크톱 응용프로그램을 만들기에 매우 적합한 자바 개발 방식이다. 그러므로 RCP 플러그인을 이용하면 실행 모듈 exe를 만들 수 있으며 시스템 트레이에서 작동하도록 응용프로그램을 구성할 수 있다.

가. UI 구현

MISO의 모든 UI는 자바로 작성된 MISO.exe에서 생

성한다. 검색의 실행 과정의 예를 보인 그림 5, 그림 6, 그림 7의 UI는 모두 MISO.exe에서 만들었다.

나. Open API 호출 및 XSLT

본 연구에서 기능 구현을 위해 사용한 사이트와 Open API 들을 표 2에 정리하였다. 사전 검색의 경우 검색어를 분석하여 영어인지 한글이 판단하고 만일 영어이면 Open API에서 krDic.naver.com 대신 endic.naver.com으로 사용한다. 또한 사전 검색의 경우 Open API의 결과물은 HTML페이지 이므로 이를 내장 브라우저를 이용하여 간단히 출력한다.

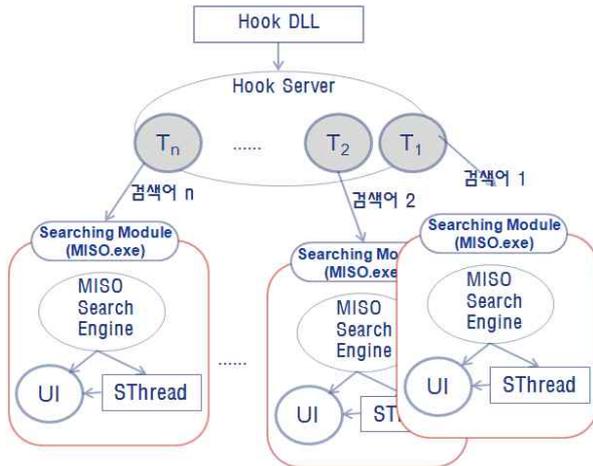
표 2. 검색 타입별 사용된 Open API
Table 2. Open APIs for each search type

검색 메뉴	사이트	OpenAPI 구성
사전	네이버	http://krDic.naver.com/small_search.nhn?query=keyword
뉴스	네이버	http://openapi.naver.com/search?key=xxx&query=keyword&target=news&start=1&display=20
백과	네이버	http://openapi.naver.com/search?key=xxx&query=keyword&target=encyc&start=1&display=10
사진	플리커	http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=xxx&text=keyword&per_page=30&page=0
동영상	유튜브	http://gdata.youtube.com/feeds/api/videos?vq=keyword&orderby=rating&start-index=0&max-results=10
번역	구글	http://www.google.com/jsapi 사이트에서 제공하는 자바스크립트의 google.language.translate() 메소드 호출

네이버 뉴스, 네이버 백과, 플리커, 유튜브 등은 모두 정상적인 REST로 작동하므로 Open API 호출에 대한 결과로서 XML을 리턴한다. 본 논문에서는 XSLT(Extensible Stylesheet Language Transformations)[13,14] 기술을 이용하여 XML 문서를 HTML 문서로 변환하였다. 이를 위해 XSL(Extensible Stylesheet Language) 문서를 각각 따로 작성하였다.

5. 검색당 하나의 MISO.exe 인스턴스 생성

하나의 검색어에 대한 검색 이벤트가 발생하면 한 번의 검색 과정이 실행된다.



*T₁, T₂, T_n는 윈도우 스레드임

그림 9. 각 검색마다 스레드에 의해 관리
Fig. 9. One thread created per each search

사용자가 여러 번의 검색을 하는 경우 MISO는 각 검색마다 독립적으로 결과를 유지하도록 한다. 이를 위해 그림 9와 같이 구현하였다. HookServer 모듈은 하나의 검색어에 대해 검색 이벤트가 발생하면 윈도우 스레드를 하나씩 만들고 이 스레드가 MISO.exe를 자식 프로세스로 생성한다. 스레드는 MISO.exe를 자식 프로세스로 생성한 뒤 단순히 프로세스가 죽기를 기다린 후 종료한다.

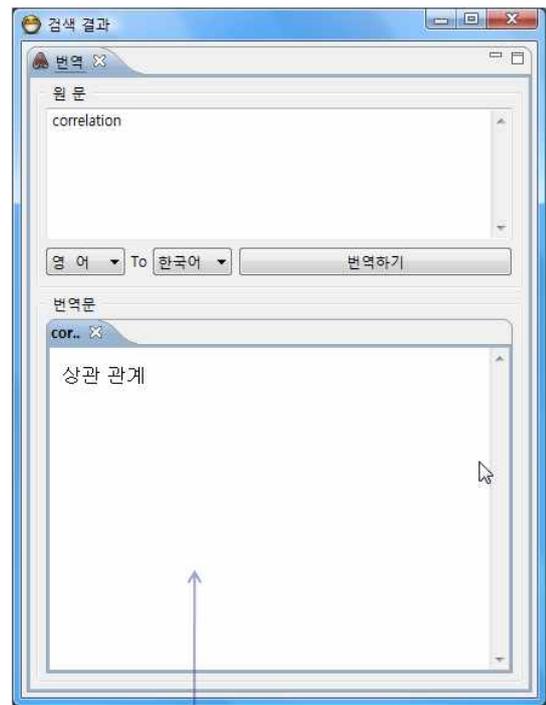
6. 번역 구현

번역 기능을 위해 Open API를 제공하고 있는 사이트는 현재 발견할 수 없기 본 연구에서는 구글 사이트에서 제공하는 번역 웹 페이지를 이용하였다.

그림 10은 구글 사이트에서 제공하는 자바스크립트코드를 이용하고 이를 활용하여 번역을 명령하는 자바스크립트를 내장한 웹 페이지를 제작하고 이 페이지를 출력하도록 하였다. 페이지가 출력되면 자동으로 구글에 번역을 지시한다. 그림 10은 현재 "correlation" 단어를 영어("en")에서 한국어("ko")로 번역하도록 구성된 예이다. 그림 11은 correlation을 검색어로 주었을 때 MISO의 번역 결과를 보여준다.

```
<html><head>
<script type="text/javascript"
src="http://www.google.com/jsapi"></script>
<script type="text/javascript">
google.load("language", "1");
function initialize() {
var text = "correlation";
google.language.translate(text, "en", "ko",
function(result) {
var translated =
document.getElementById("translation");
if (result.translation) {
translated.innerHTML = result.translation;
}
});
google.setOnLoadCallback(initialize);
</script></head>
<body>
<div id="translation"></div>
</body>
</html>
```

그림 10. 구글의 번역 기능을 사용하기 위한 HTML 문서
Fig. 10. HTML text for using Google's interpreting function



구글의 번역 기능을 사용하기 위한 HTML이 있는 콘트를

그림 11. MISO의 번역 기능 사례
Fig. 11. Example of Interpreting in MISO

V. 결론

본 논문은 컴퓨터 상에서 문서 작업 도중 간편하게 필요한 검색을 수행할 수 있는 매쉬업 소프트웨어의 설계 및 구현 사례를 소개하였다. 본 논문의 검색 방법은 기존의 웹 검색 시스템은 웹 브라우저를 따로 실행하고 여러 페이지의 방문을 거쳐 검색어를 입력하는 수고를 더는 방법이다. 어떤 소프트웨어를 사용 중이더라도 화면에서 검색어를 선택하고 CTRL키와 마우스왼쪽 버튼을 동시에 누르면 바로 검색이 일어난다. 또한 검색 엔진을 탑재하지 않고 검색 정보를 제공하는 여러 사이트의 Open API를 이용하기 때문에 동시에 여러 종류의 서로 다른 검색을 실행할 수 있으며 지도 검색 등 새로운 검색 타입이 필요하면 그에 해당하는 Open API만 호출하면 되므로 기능 면에서 확장성이 뛰어나다.

본 논문의 설계 및 구현 사례는 매쉬업을 활용하는 개발에 좋은 참고 사례가 될 것이다. 추후 지도 검색 등 검색 타입을 보다 더 추가하여 충분한 기능을 갖춘 검색 소프트웨어로 업그레이드하고자 한다.

참 고 문 헌

- [1] <http://www.google.com>
- [2] <http://www.yahoo.com>
- [3] <http://www.naver.com>
- [4] <http://www.daum.net>
- [5] <http://www.flicker.com>
- [6] 박지강, 당신은 웹 2.0 개발자입니까?, 한빛미디어, 2007
- [7] Sunilkumar Peenikal, "Mashups and the Enterprize", White Paper, Mphasis, Sep. 2009
- [8] John Battelle, The Search:How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture, Portfolio Hardcover, 2005
- [9] <http://www.logitech.com>
- [10] <http://kr.dic.yahoo.cm>

- [11] Roy Fielding, "Architectural Styles and the Design of Network-based Software Architectures", dissertation, Univ. of California, Irvine, 2000.
- [12] <http://www.eclipse.org>
- [13] XSL Transformations Version 2.0, <http://www.w3.org/TR/xslt20>
- [14] Kay, Michael, XSLT 2.0 and XPath 2.0 Programmer's Reference, Wiley Press

저자 소개

황 기 태(정회원)



- 서울대학교 컴퓨터공학과 학사
- 서울대학교 컴퓨터공학과 석사
- 서울대학교 컴퓨터공학과 박사
- 경력
- University of Callifornia, Irvine 방문교수

<주관심분야 : 모바일 & 유비쿼터스 시스템, 콘텐츠 스트리밍>

※ 본 연구는 한성대학교 교내 연구비를 지원받았음.