효율적인 검색 인터페이스를 위한 웹 기반 컴퓨터 용어사전의 설계 및 구현

황 병 연[†] · 박 성 철[††]

요 약

본 논문에서는 인터넷을 통해 실시간으로 항상 최신의 컴퓨터 용어 검색 서비스를 제공할 수 있는 웹을 기반으로 한 컴퓨터 용어 사전을 설계하고 구현하였다. 본 용어사전은 FOLDOC(Free On-Line Dictionary Of Computing)의 사전을 기본으로 영문 해설을 제공하고 각 용어에 대해 한 명 이상의 번역자가 번역할 수 있도록 함으로써 기존 컴퓨터 사전에서 제공하지 않는 기능을 추가하였다. 그리고 SQL Server DBMS와 SQL을 이용한 다양한 검색 인터페이스(입력 문자로 시작하는 용어 검색, 입력 문자가 해설에 들어간 용어 검색 등)를 제공함으로써 적은 정보만으로도 원하는 용어를 검색할 수 있게 하였다. 본 컴퓨터 용어 사전의 성능 평가를 위해서 FOLDOC Mirror Site의 로그를 분석하여 CPU 부하율을 측정하였다. 실험 결과 본 컴퓨터 용어 사전은 최대 1780여명 이상의 동시 사용자를 수용할 수 있다는 결론을 얻었다.

Design and Implementation of Web-Based Dictionary of Computing for Efficient Search Interface

Byung Yeon Hwang[†] · Sung-cheol Park[††]

ABSTRACT

In this paper, we designed and implemented a web-based dictionary of computing which keeps the data up-to-date. This dictionary shows the English information based on the FOLDOC (Free On-Line Dictionary Of Computing) dictionary file at the beginning of searching, and then one or more users can translate the information into Korean. This function is the new one only this dictionary has. Also, we can easily find any words we want to look up, even if we don't know the spelling completely, because the dictionary has various searching interfaces (searching for the words starting with inputted characters, searching for the words including inputted characters in the description, etc.) using a SQL Server DBMS and SQL. The performance test for CPU load factor shows that the server can support at least 1780 users at the same time.

키워드 : 컴퓨터 용어 사전(Computing Dictionary), 웹 기반(Web-based), 데이터베이스 관리 시스템(DBMS), 매칭 에이전트(Matching Agent), 한글 번역(Hangul Translatation), 자바 서버 페이지(JSP), 자바(Java), 자바 데이터베이스 연결(JDBC)

1. 서 론

컴퓨터 산업의 급속한 발전으로 수많은 컴퓨터 용어들이 생겨나고 있으며 같은 용어라도 시기나 지역에 따라 다른 의미를 가지는 경우도 많다. 그래서 컴퓨터 용어 사전의 경우 다른 사전류 보다 자주 새 용어를 추가하거나 기존 용어를 갱신해야 한다. 컴퓨터 용어 사전을 책의 형태로 출판할 경우 시간이 지남에 따라 가치가 급속히 떨어지게 된다. 이 때문에 항상 최신의 정보를 제공할 수 있도록 인터넷을 통해서 컴퓨터 용어 사전을 서비스할 필요가 있다.

본 논문에서는 위의 조건을 만족할 수 있도록 FOLDOC (Free On-Line Dictionary Of Computing)[1]의 사전을 이용해서 웹을 기반으로 한 컴퓨터 용어 사전을 설계하고 구현한 다음 성능 평가를 하였다. 수시로 갱신이 되는 FOLDOC의 사전을 데이터베이스화해서 영문으로 우선 용어 해설을 서비스하고 한 명 이상의 번역자가 한글 해설을 입력하게 함으로써 한국적 용어 정의를 유도하였다. 뿐만 아니라 다양한 검색 인터페이스를 제공하여 적은 정보만으로도 쉽게 원하는 용어를 찾을 수 있도록 하였다. 그리고 성능평가를 위해서 본 연구실에서 운영중인 FOLDOC Mirror Site의 로그파일을 분석하여 성능 평가 기준을 세우고 동시 사용자 수에 따른 CPU 부하율을 중심으로 실험하였다.

본 논문의 구성은 다음과 같다. 2장에서는 웹을 기반으로 한 기존의 컴퓨터 용어사전과 본 논문에서 구현한 용어사전을 사용자 인터페이스의 다양함을 기준으로 비교를 하였다. 3장에

※ 본 연구는 2002년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.
† 종신회원 : 가톨릭대학교 컴퓨터전자공학부 교수
†† 준 회 원 : 포항공과대학교 컴퓨터공학과
논문접수 : 2001년 4월 16일, 심사완료 : 2001년 7월 20일

서는 본 컴퓨터 용어사전의 구조를 나타내었다. 4장에서는 본 컴퓨터 용어사전의 주요 모듈에 따라서 독창적으로 개발한 구현 방법들에 대해 설명하였다. 5장에서는 본 컴퓨터 용어사전의 주요화면을 중심으로 제공하는 서비스를 소개하였다. 6장에서는 본 컴퓨터 용어 사전의 성능을 평가하였다. 끝으로 7장에서 결론을 맺는다.

2. 기존 컴퓨터 용어 사전과의 비교

본 컴퓨터 용어 사전은 타 용어 사전이 제공하지 않는 특징적인 기능이 3가지가 있는데, 입력문자로 시작하는 용어 검색과 입력문자가 해설에 들어간 용어 검색기능 그리고 한글화 기능이 바로 그것이다.

입력 문자로 시작하는 용어 검색과 입력문자가 들어간 용어 검색은 검색하고자하는 용어의 정확한 스펠링을 모를 때 사용할 수 있는 기능이다. 전체 스펠링은 모르더라도 첫 부분의 스펠링은 알 경우에 입력문자로 시작하는 용어검색 기능을 이용하면 입력문자가 들어간 용어 검색 기능을 사용했을 때보다 훨씬 정확도 높은 용어들만으로 추출할 수 있다. 입력문자가 들어간 용어 검색 기능은 일부 스펠링은 아는데 시작부분이 아니거나 입력 용어가 포함된 용어를 찾을 때 이용할 수 있다.

입력 문자가 해설에 들어간 용어 검색 기능은 입력 문자가 해설에 포함되어 있는 용어를 검색하는 기능이다. 이 기능을 사용하면 카테고리의 영향을 받지 않고 특정 주제와 관련된 용어를 찾을 수 있다. 경우에 따라서는 카테고리 검색보다 정확도도 높고 검색 범위도 많이 줄어든다.

한글화 기능은 영어로 된 용어를 한글로 정의해서 입력할 수 있게 하는 기능이다. 컴퓨터 용어는 특성상 영어로 명명된 외국어가 대부분이므로, 컴퓨터의 빠른 발전 속도에 발맞추기 위해서는 외국어로 정의된 용어를 신속하게 한글화하는 것이 관건이다. 본 컴퓨터 용어 사전은 수시로 새 용어를 추가하고 기존에 정의된 용어도 수정하는 FOLDOC 사전을 이용해서 우선적으로 영어로 사전 서비스를 제공하고, 용어 검색과 한글로 번역하는 작업을 같은 페이지에서 할 수 있도록 함으로써 신속하게 번역할 수 있게 했다. 본 컴퓨터 용어 사전에서는 영문해설을 참고하여 한글로 한국의 실정에 맞도록 독자적으로 재 정의할 수 있다. 또한 한 용어에 대해 복수개의 번역문을 입력하고 입력된 번역문을 수정하거나 삭제할 수 있게 되어있다. 이렇게 함으로써 다른 사전과는 달리 다양한 관점에서의 정의를 유도하고 한국의 상황에 맞도록 용어를 많은 사람들이 수정할 수 있도록 하였다. 한글로 입력할 때에는 입력자의 이름, 소속, 메일 주소를 기입하도록 함으로써 입력자가 되도록 신중을 기하도록 유도하였다. 그리고 각 해설에 점수를 매김으로써 한 용어에 대해 여러 개의 해설이 존재할 경우 우수한 해설을 식별하기 쉽도록 하였다.

이 외에 중점을 두어 볼만한 사항은 본 컴퓨터 용어 사전에서는 통계 처리를 통한 추가 서비스도 제공을 하고 있고 타 컴퓨터 용어 사전에 비해 다양한 추가 서비스를 제공할 수 있는 여건을 갖추어 놓았다는 점이다. 현재 본 컴퓨터 용어 사전에서는 각 검색 용어에 대해 해당 용어를 검색한 사람들이 당일 하루동안 검색한 용어들을 찾아준다. 이 용어들을 통해 자신과 비슷한 관심을 가지고 있는 사람들이 다른 어떤 용어들에 관심을 가지고 있는지를 알 수 있게 함으로써 매칭 에이전트 개념을 추가하였다[2, 3].

본 논문에서는 위의 기능 등을 구현하기 위해서 하나의 파일로 되어있는 FOLDOC의 사전을 데이터베이스화하였고 번역문도 데이터베이스에 저장하도록 하였다. 또한 사용자가 용어를 검색할 때마다 검색 내역인 검색 용어 명, 사용자 컴퓨터의 IP 주소, 검색한 날짜도 데이터베이스에 기록함으로

〈표 1〉 주요 컴퓨터 용어 사전과 본 컴퓨터 용어 사전의 검색 기능별 비교

사전 / 항목	텀즈 코리아 [4]	돌도끼 [5]	아이누리 컴퓨터대사전 [6]	EzWord [7]	Webopedia [8]	what?is [9]	FOLDOC	가톨릭대 컴퓨터 용어사전 [10]
A				V	V	V	V	V
B	V					V	V	V
C		V	V		V	V	V	V
D								V
E		V						V
F								V
G	V				V	V	V	V
H	V				V	V	V	V
I							V	V
J	V	V	V		V	V	V	V
K					V	V	V	V
L	V				V	V		V

A : 카테고리별 검색, B : 알파벳 순 검색, C : 입력용어 검색, D : 입력문자로 시작하는 용어 검색, E : 입력문자가 들어간 용어 검색, F : 입력문자가 해설에 들어간 용어 검색, G : 검색용어의 설명에 포함된 컴퓨터 용어 검색, H : 검색 용어와 관련된 HTTP 사이트 링크, I : 검색 용어와 관련된 FTP 사이트 링크, J : 설명화면과 같은 화면에서 용어 검색, K : 검색한 용어와 같은 카테고리에 속하는 용어 검색, L : 검색 통계를 이용한 용어 검색

써 통계 기능을 지원할 수 있도록 하였다.

본 컴퓨터 용어 사전에서는 총 9가지의 용어 및 카테고리 검색 기능을 제공하고 있다. 이를 위해서 여러 개의 SQL문을 이용한 다양한 검색 모듈들이 연결되어 있다. 이렇게 함으로써 각 사용자에게 적합한 검색 서비스를 제공한다.

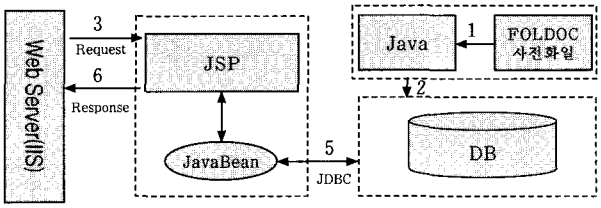
한글로 번역을 간편하게 하기 위해서 영어검색 화면과 같은 페이지에 번역문 입력 폼을 만들었다. 번역문을 저장할 때에는 필수 항목을 비워두지는 않았는지 확인한 후에 데이터베이스에 기록한다. 번역문을 검색할 때에는 해당 용어에 대한 번역문 모두를 같은 페이지에 나타내도록 하였으며 각 번역문의 점수를 매기기 위해 추천 기능을 포함시켰다.

인터넷을 통해 서비스하는 컴퓨터 용어 사전의 경우에는 사용자 인터페이스의 다양함과 편리함도 매우 중요하다. <표 1>은 본 논문에서 구현한 사전과 기존 사전의 검색 기능을 비교한 것이다.

3. 시스템 구조

3.1 시스템 구조

본 컴퓨터 용어사전은 Windows NT와 관계형 데이터베이스 관리 시스템인 MS-SQL Server를 기반으로 하여 Java와 JavaBeans 그리고 JSP(JavaServer Pages™)[11]를 사용하였다[12]. 웹 서버로는 IIS(Internet Information Server)를 사용하였다. (그림 1)은 시스템의 구조를 나타내고 있다.



(그림 1) 시스템 구조

(1) FOLDOC 사전 파일에서 용어 스트링을 읽는다.
(2) 읽어들인 스트링을 분석해서 용어 명, 카테고리, 해설로 분류해서 DB에 저장한다.
(3) 사용자가 웹브라우저를 통해 검색을 요청하면 웹서버는 JSP로 입력정보를 넘겨준다.
(4) JSP는 검색 요구에 따라 해당 JavaBean에 정보를 요청한다[13, 14].
(5) JavaBean은 JDBC를 통해 DB에 검색 SQL문을 보내거나 저장 프로시저(stored procedure)를 실행시킨다[15, 16]. DB에서 검색 결과를 전달해주면 이 결과를 정리해서 JSP로 넘겨준다.
(6) JSP는 JavaBean으로부터 넘겨받은 검색 결과를 사용자 인터페이스 형식에 맞게 구성해서 Web Server로 보내준다. 그러면 Web Server는 사용자 웹브라우저로 결과를 전송한다.

3.2 기능별 구성 요소

본 컴퓨터 용어 사전은 크게 네 부분으로 구성된다. FOLDOC 사전 파일 데이터베이스화 모듈, 데이터베이스에 저장된 용어 검색 모듈, 한글화 모듈, 검색 통계 모듈로 나뉜다. (그림 2)는 이 구성 요소간의 관계를 보여준다.

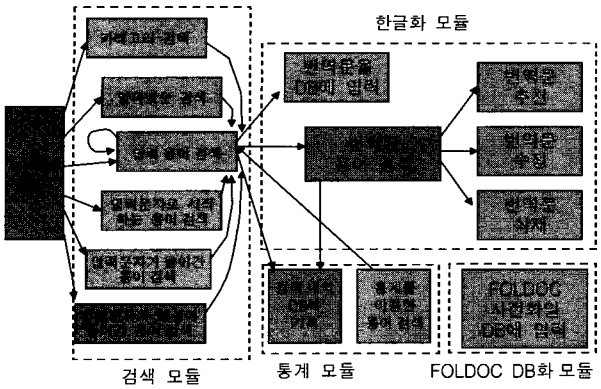
FOLDOC 사전파일 데이터베이스화 모듈은 FOLDOC의 사전 파일을 데이터베이스에 입력하는 모듈이다. 이 모듈은 사전 서비스 시작 전에 FOLDOC의 사전 파일 전체를 데이터베이스에 입력시키거나, FOLDOC 사전이 업데이트 되었을 경우 업데이트 된 부분을 데이터베이스에 추가시킬 때에 사용한다.

검색 모듈은 카테고리 별 검색, 알파벳 순 검색, 입력 용어 검색, 입력문자로 시작하는 용어 검색, 입력문자가 들어간 용어 검색, 입력문자가 해설에 들어간 용어 검색으로 구성된다. 입력 용어 검색을 제외한 나머지 검색은 찾은 용어 리스트를 보여준다. 이 리스트에서 한 용어를 선택하면 선택된 용어 명을 입력 용어 검색 모듈로 넘겨서 해당 용어에 대한 세부 내용을 보여준다. 여기서 알 수 있듯이 입력 용어 검색은 입력된 용어 하나를 검색하고 세부 내용을 보여준다. 세부 내용에서는 검색 용어의 번역문 검색, 해당 카테고리에 속하는 용어 검색, 해설에 포함된 컴퓨터 용어 검색을 할 수 있다. 기본적으로 영어로 세부 내용이 나오고 해당 용어가 번역되어 있을 때에는 번역문을 볼 수 있다.

한글화 모듈은 입력 용어 검색 기능과 연결되어있다. 용어를 검색해서 하나의 용어에 대한 세부 내용이 나온 페이지의 아래쪽에는 검색된 내용을 한글로 입력할 수 있는 폼이 있다. 이 폼에 각 항목을 입력하면 입력 항목들을 데이터베이스에 입력한다. 번역된 내용을 보기 위해서는 영어로 해당용어를 검색한 다음 번역문 검색을 하면 된다. 한 용어에 대해 복수개의 번역문이 있을 수 있으며, 각 번역문은 비밀 번호가 맞을 경우에 한해서 수정 및 삭제가 가능하다. 그리고 검색자는 특정 번역문을 추천할 수 있다.

검색 통계 모듈은 크게 검색 내역을 입력하는 기능과 통계를 이용해서 용어를 검색하는 기능으로 나눌 수 있다. 입력하는 기능은 검색한 용어, 사용자 컴퓨터의 IP 주소, 검색한 날짜를 데이터베이스에 저장한다. 검색한 용어는 영문으로 검색했을 경우와 번역문을 검색했을 경우 서로 다른 테이블에 저장된다. 통계를 이용한 검색 기능은 저장된 정보를 바탕으로 많은 통계치 또는 용어들을 검색할 수 있다. 현재 본 컴퓨터 용어 사전에서는 검색요청을 한 용어를 이미 검색한 다른

사람들이 당일 검색한 용어들을 찾아준다.



(그림 2) 기능별 구성 요소

4. 주요 모듈의 구현

사용자의 검색 요구를 처리할 때 가장 시간이 오래 걸리는 부분이 데이터베이스와의 커넥션을 생성할 때이고 그 다음이 질의문을 전송하고 받을 때이다. 그래서 본 컴퓨터 용어사전에서는 커넥션을 미리 만들어놓고 필요할 때 사용하는 DBPool이라는 기법을 썼다. 그리고 질의문 전송을 줄이기 위해서 통계 모듈에서는 저장 프로시저(stored procedure)를 호출하여 한 번의 질의문 전송으로 해결했으며 번역 부분에서는 엔터티 삽입시 중복 확인과 비밀번호 확인을 위한 질의문 전송을 없애는 기법을 개발해서 사용했다.

본 컴퓨터 용어사전에는 기본적으로 8개의 테이블이 사용되었다. FOLDOC의 사전 내용을 저장하는 영문 기본 테이블인 comdic_eng 테이블, 한글로 번역했거나 새로 정의한 내용을 저장하는 한글 기본 테이블인 comdic_kor 테이블, 영어 용어명과 거기에 대응되는 한글 용어 명을 저장하는 comdic_eng_kor_match 테이블, 영어 카테고리 명에 해당하는 한글 카테고리 명을 저장하고 있는 category_eng_kor_match 테이블, 영어 용어를 검색했을 때 검색 내역을 기록해두는 comdic_ratings_eng 테이블, 한글 용어를 검색했을 때 검색 내역을 기록해두는 comdic_ratings_kor 테이블, 영어 용어의 검색 횟수를 기억하고 있는 comdic_eng_ratings 테이블 그리고 한글 용어의 검색 횟수를 기억하고 있는 comdic_kor_ratings 테이블이 사용되었다.

4.1 FOLDOC 사전 파일 데이터베이스화 모듈

FOLDOC은 사전이 하나의 파일로 되어 있다. FOLDOC의 웹 페이지에서 사용자가 검색 용어를 입력하면 해당 용어와 같은 용어를 keys파일에서 찾고 이 keys파일에서 찾은 용어에 해당하는 index offset을 offset파일에서 찾는다. 찾은 offset 범위의 string을 사전 파일에서 읽어서 검색 자에게 제공하게 된다. FOLDOC의 사전파일에서 각 용어는 용어명, 카테고리, 용어해설 이렇게 세 부분으로 구성되어 있다. 그래서 데이터베이스의 영어 용어 테이블(comdic_eng 테이블)에 이렇게 세 어트리뷰트로 입력하였다.

FOLDOC사전의 경우에는 offset을 이용하기 때문에 순서화의 제약을 받는다. 즉 사전의 용어를 추가하거나 수정, 삭제하기가 어렵다. 반면에 본 컴퓨터 용어 사전의 comdic_eng 테이블에서는 id를 넣지 않고 용어명을 기본 키로 설정함으로써 추후 사전 업데이트 및 수정을 용이하게 하였다.

4.2 용어 검색 모듈

용어 검색 모듈에서는 변수 값을 다른 모듈 또는 데이터베이스에 넘겨주거나 HTML형식으로 웹브라우저에 나타내기 위해서 각각의 경우에 특별한 처리가 필요하다.

각 검색 기능들간에 검색하고 있는 용어에 대한 변수 값을 넘겨줄 때에는 CGI방식에서의 POST형식 또는 GET형식으로 전달하였다. GET방식에서는 '+', '&', '#' 등을 특별한 의미로 사용하고 있다. 그래서 전달하는 변수 값에 이 기호가 있을 경우에는 CGI 예약어가 아닌 전달 문자로 인식할 수 있게 바꾸어 주어야 한다. 변수를 SQL 질의문에 포함시킬 경우에도 홑 따옴표는 SQL 예약어이므로 예약어가 아닌 기호로 인식할 수 있게 해야 한다. 또한 변수 값에 '<', '>', '\n', ' ', '\t', '\n ', '\n\t'가 있으면 HTML에서는 해당 부분을 나타내지 않으므로 따로 처리를 해 주어야 한다.

4.3 한글화 모듈

한글화 모듈은 번역문 입력 기능, 번역된 용어 검색 기능, 삭제 기능, 수정 기능, 추천 기능 등으로 구성되어있다. 이 중 번역된 용어 검색 기능과 추천 기능은 기존의 구현 방식과 같으므로 본 절에서는 나머지 기능에 대해서만 설명한다. 번역문 입력 기능의 알고리즘은 다음과 같다.

```
TranslationInsert(word_eng, word_kor, name, email, position, student_num, translation, password)
/*word_eng : 영문 용어명, word_kor : 사용자가 정의한 한글 용어명, name : 번역자 이름, email : 번역자 메일주소, position : 소속, student_num : 학생번호, translation : 번역문 또는 해설, password : 비밀번호*/
{RequiredCheck(name, word_kor, translation) ; /*필수 입력사항 확인*/
 If(word_kor_id == -1) /*word_kor_id : 새 한글 용어 ID번호*/
    word_kor_id = GetMaxId( ) ; /*번역문을 입력하는 comdic_kor 테이블은 word_kor와 word_kor_id가 기본키이다. JSP/Servlet 엔진 가동 후 처음 번역할 때 GetMaxId( )로 새 word_kor_id 값을 구한 다음 scope가 application인 변수인 word_kor_id에 저장해 놓고 JSP/Servlet 엔진이 꺼질 때까지 이 변수를 사용한다. 이렇게 함으로써 질의문 전송을 많이 줄일 수 있다.*/
 date = TodayDate( ) ; /*입력 날짜*/
 InsertIntoComdic_kor(word_kor, word_kor_id, name, email, position, student_num, translation, password, date) ; /*comdic_kor 테이블에 입력*/
```

```
InsertIntoComdic_eng_kor_match(word_eng, word_kor) ;
  /*comdic_eng_kor_match 테이블(용어 명 매칭 테이블)에 word_eng
와 word_kor를 삽입한다. 기본키가 word_eng와 word_kor이므로 중
복 확인을 하는 질의문을 전송할 필요 없이 입력 질의문을 데이터베이
스에 전송하면 데이터베이스 관리 시스템 자체에서 중복이 안될 경우
에만 테이블에 삽입한다.*/
}
```

삭제 기능, 수정 기능에서도 위의 InsertIntoComdic_eng_kor_match( )의 기법을 사용해서 새 한글 용어명의 중복 확인과 비밀번호 확인을 위한 질의문 전송을 없앴다. 번역된 한글 용어명을 수정할 경우에는 중복확인을 하고 테이블에 같은 레코드가 없을 경우에는 용어 명 매칭 테이블(comdic_eng_kor_match)에 삽입해야 하는데 입력모듈에서와 마찬가지로 확인 질의문 처리 없이 바로 데이터베이스에 입력 질의문을 전송한다. 비밀번호 확인도 질의문 하나에 포함시켰다. 다음은 삭제 기능에 이 기법을 적용한 예를 나타낸다.

```
String Delete( )
{     deleteQuery = "delete
                    from comdic_kor
                    where word_kor = "+deleteWord_kor+"
                      and word_kor_id = "+deleteWord_korId+"
                      and password = "+inputtedPassword ;
      ...
      int deletedRows = stmt.executeUpdate(deleteQuery) ;
      if(deletedRows>=1)
      {       return "Deleted" ;
      }else   return "Not Deleted" ;
}
```

위에서 word_kor과 word_kor_id는 번역문 저장 테이블의 기본키이다. 그래서 이 두 값을 주면 하나의 레코드만 선택된다. 여기에 password 어트리뷰트도 같아야 삭제가 되므로 비밀번호가 같을 경우에만 삭제가 된다. 그리고 deletedRows는 질의문 전송 후 삭제된 레코드의 수를 넘겨받게 되는데 이 삭제된 레코드의 수로 비밀번호가 맞았는지를 확인할 수 있다. 즉, 삭제된 레코드 수가 0개이면 비밀번호가 틀린 것이고 1개이면 비밀번호가 맞는 것이다. 다음은 수정 기능에 대한 적용 예를 나타낸다.

```
Modify( )
{
  modifyQuery = "update comdic_kor
                set name ="+ newName +"
                ...
                where word_kor_id = '"+ modifyWord_korId +"'
                  and password = "+ inputtedPassword ;
  int modifiedRows = stmt.executeUpdate(modifyQuery) ;
  if(modifiedRows >= 1)
  {
    NewKorQuery = "insert into comdic_eng_kor_match (word_eng,
                          word_kor)
                          values('"+ modifyWord_eng +"',
                          '"+ modifyWord_kor +"')" ;
        stmt.executeUpdate(NewKorQuery) ;
  }
}
```

위의 수정에서도 Delete( )에서와 같은 방법으로 비밀번호를 확인했고 비밀번호가 맞을 경우에는 영어 용어명 매칭 테이블에 추가 중복확인 질의문 전송없이 새 용어명을 삽입했다.

4.4 검색 통계 모듈

검색 통계는 사용자가 검색할 때마다 여러 개의 질의문을 전송해서 데이터베이스를 갱신해야한다. 이 부분을 저장 프로시저로 만들어놓음으로써 처리 속도를 크게 향상시켰다. 다음은 영문 검색용 프로시저와 번역문 검색용 프로시저 중 영문 검색용을 나타낸다.

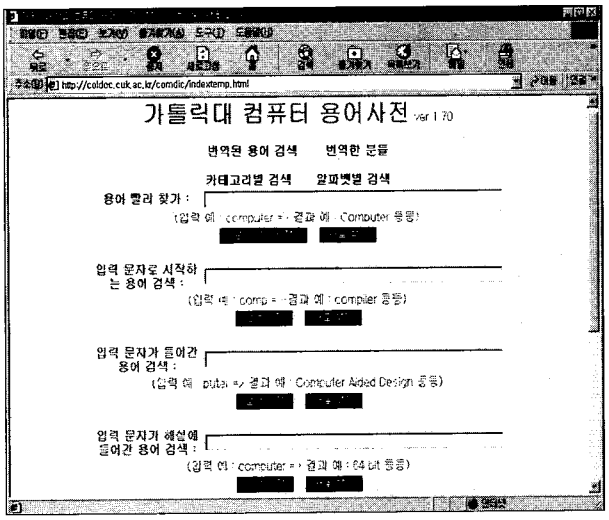
```
CREATE PROCEDURE RecodeEngSearch @word_eng varchar(255),
@user_ip varchar(50), @today_date int AS /*@word_eng : 검색 용
어명, @user_ip : 사용자IP, @today_date : 검색한 날짜*/
begin transaction
  insert into comdic_ratings_eng(word_eng, user_ip, search_date)
  values(@word_eng, @user_ip, @today_date) /*검색 내역을 삽입*/

  declare @existence_check int
  select @existence_check = count(word_eng) from comdic_eng_
  ratings where word_eng=@word_eng /*해당 용어의 검색 횟수를
  기록한 레코드의 존재 확인*/
  if @existence_check =0 /*해당 용어를 이전에 검색한 적이 없는 경우*/
    begin
    insert into comdic_eng_ratings(word_eng, rating) values(@word
    _eng,1)
    end
  else /*해당 용어를 이전에 검색한 적이 있는 경우. 검색 횟수를 1
       증가시킴.*/
    begin
    update comdic_eng_ratings set rating=rating+1 where word_
    eng=@word_eng
    end
  commit transaction
```

위의 저장 프로시저에서 날짜는 정수형이다. 날짜를 날짜형이 아닌 정수형으로 만들면 년, 월, 일의 세 변수로 분리하지 않고도 Boolean Logic을 사용해서 간편하게 계산할 수 있는 이점이 있다. 예를 들어 2001년 5월 1일부터 2001년 10월 1일 이전까지를 검색하려면 날짜 값이 20010501보다 크거나 같고 20011001보다 작은 투플들을 고르면 된다.

사용자의 검색 사항 등을 저장해 놓으면 추후에 다양한 부가 서비스를 제공할 수 있다. 한 예로 사용자 컴퓨터의 IP주소를 이용해서 별도의 회원 가입 없이도 개인별 사전을 제공할 수 있다. 개인별 사전을 이용하면 개인별 저장 용어를 검색할 수 있다. 그리고 각 용어에 대해 해당 카테고리와 해당 용어를 검색한 사람들의 검색 이력을 이용하여 관련성이 높은 용어를 찾아줄 수도 있다. 이 외에도 전체 사용자의 검색 현황을 주기적으로 통계를 내어서 시기별 검색 추이를 정리할 수도 있고 다양한 서비스를 손쉽게 개발할 수 있다.

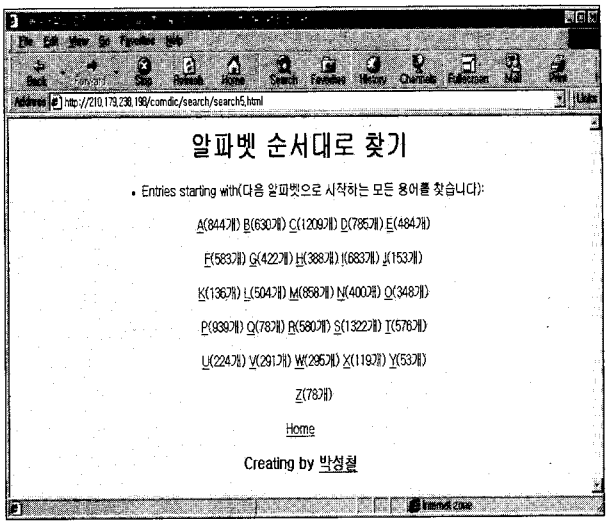
5. 컴퓨터 용어 사전의 서비스 예

본 장에서는 제안된 컴퓨터 용어 사전의 가장 중요한 기능인 입력 용어 검색 기능을 포함한 여러 기능을 검증하는 예를 보인다. (그림 3)은 인덱스 화면을 나타낸다. 여기서 카테고리 검색, 알파벳 순 검색, 입력 용어 검색, 입력문자로 시작하는 용어 검색, 입력문자가 들어간 용어 검색, 입력문자가 해설에 들어간 검색 중 하나의 방법으로 검색할 수 있다.

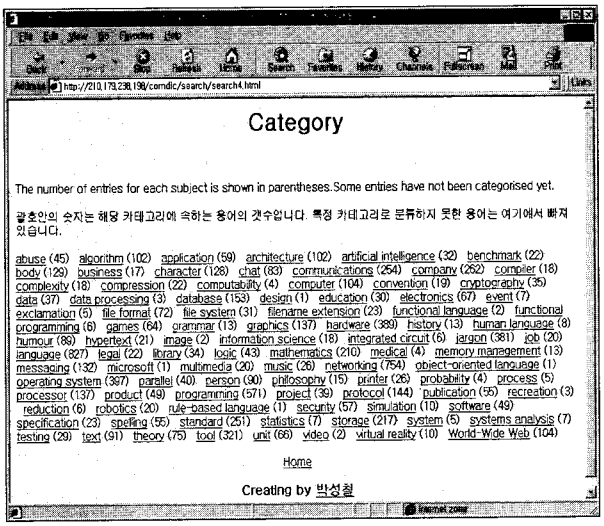


(그림 3) index 화면

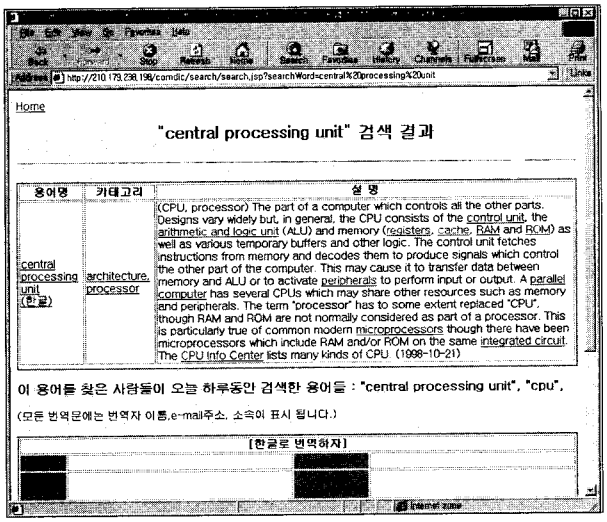
(그림 4)는 알파벳순으로 검색하는 화면이다. 각 알파벳으로 시작하는 용어들을 찾을 수 있다. (그림 5)는 카테고리 별로 검색하는 화면이다. 각 카테고리에 속하는 용어들을 검색할 수 있다. 용어에 따라서는 아무 카테고리에도 속하지 않는 경우도 있는데 이러한 용어들은 여기에서 검색할 수 없다. (그림 6)은 입력 용어명으로 'central processing unit'를 입력하여 검색한 화면이다. 용어명을 클릭하면 해당 용어의 한글 설명을 볼 수 있고, 카테고리를 클릭하면 같은 카테고리에 속한 용어 목록을 볼 수 있다.



(그림 4) 알파벳순 검색

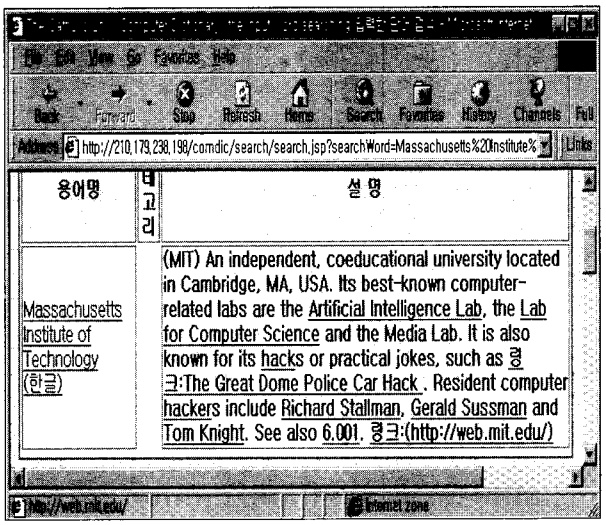


(그림 5) 카테고리 별 검색



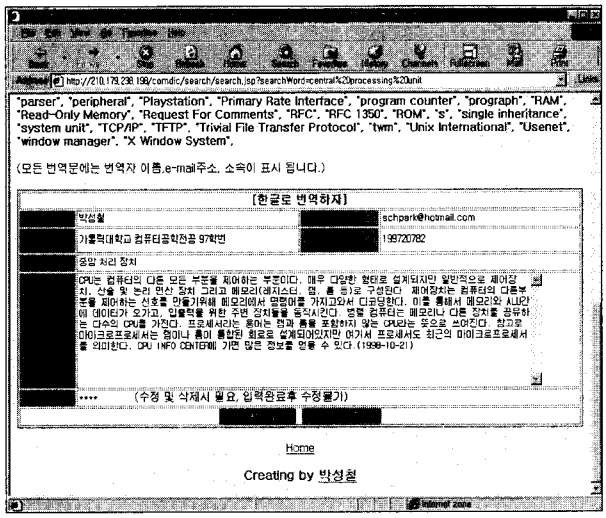
(그림 6) 입력 또는 링크로 검색된 용어

(그림 7)은 검색 용어와 관련된 HTTP 사이트로 이동할 수 있도록 링크되어 있는 화면이다. HTTP사이트나 FTP사이트와 연결된 링크 앞에는 '링크 : ' 표시를 하였다. 본 사전에서 사이트 이름을 알고 있을 경우에는 사이트 이름을 나타내었고 모를 경우에는 괄호 안에 주소를 나타내었다. (그림 7)에서 마지막 링크에 마우스 포인트를 위치시키면 화면의 왼쪽 아래에 해당 사이트로 이동할 수 있다는 표시로 해당 사이트의 주소가 나타난다.



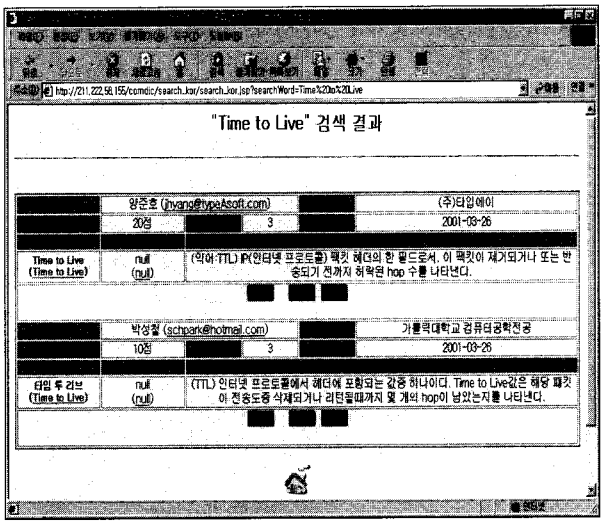
(그림 7) HTTP 사이트 링크

(그림 8)은 (그림 6)에서 영문 용어 해설을 한글로 번역해서 입력하는 화면이다.

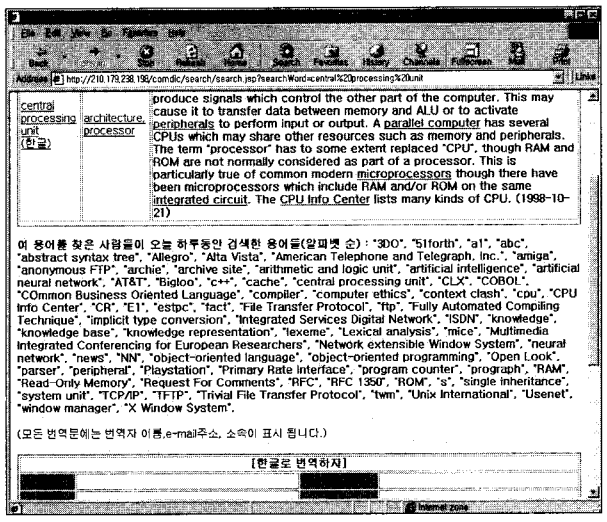


(그림 8) 번역문 입력 화면

(그림 9)는 한글 용어 해설을 검색한 화면이다. 두 개 이상의 해설이 있을 경우 한 화면에 모두 나타낸다. 각 한글 해설에는 점수가 주어지는데 검색자가 추천을 하면 점수가 증가한다. (그림 10)은 통계처리로 얻어진 용어들을 보여주는 화면이다.



(그림 9) 한글 용어 해설을 검색한 화면



(그림 10) 통계 데이터를 이용해 관련 용어를 보여주는 화면

6. 성능 평가

본 컴퓨터 용어 사전에서는 JDBC를 Native-protocol pure Java Driver로 선택함으로써 데이터베이스와의 전송에 효율성을 높였다. 그리고 DBpool 기법을 사용해서 가장 시간이 많이 걸리는 데이터베이스와의 커넥션을 미리 여러 개 만들어 놓고 사용자 연결 시에는 미리 만들어 놓은 커넥션을 할당하고 종료 시에 회수함으로써 본 사전의 반응속도를 높였다. 그리고, 두 번째로 시간이 오래 걸리는 질의문 전송에 있어서도 한 번에 여러 개의 질의를 전송해야 하는 경우에는 데이터베이스 내에 저장 프로시저를 만들어 두고 이것을 실행시킴으로써 질의문 전송 횟수를 최

소한으로 줄였다.

본 논문에서는 이러한 기법을 사용한 본 사전의 성능을 평가하기 위해서 동시 사용자 수에 따른 CPU 부하율을 이용하였다. 즉, CPU 부하율이 100%에 이르는데 필요한 동시 사용자 수를 구함으로써 실질적인 서비스 성능을 평가하였다.

6.1 성능 평가 환경

성능 평가 환경은 펜티엄Ⅱ-350 PC(RAM 64M)를 서버로 하고 같은 성능의 PC 두 대를 클라이언트로 하였다. 다양한 인터넷 회선에서의 평가를 할 수 없기 때문에 LAN환경을 이용하였다. 동시 사용자 수는 임의의 시각에 본 서버에 접속해서 검색 서비스를 이용하고 있는 사용자의 인원수라고 정의하였다.

6.2 성능 평가를 위한 가정

본 성능 평가 실험에서는 동시 사용자 수와 CPU 부하율과의 관계를 다음과 같이 구하였다. 임의의 한 사용자가 본 서버에 검색 명령을 내린 후 그 다음 검색 명령을 내릴 때까지의 시간을 N이라 하였다. 그리고 본 서버가 N시간당 M개의 검색 명령을 받으면 동시 사용자 수는 M 명이라고 정의하였다.

본 논문에서는 N을 구하기 위해 본 연구실에서 2000년 3월부터 운영해오고 있는 FOLDOC Mirror Site의 로그파일을 이용하였는데 이 로그파일에서 임의의 2일 동안의 로그 기록을 대상으로 하였다. 이 로그파일에서 클라이언트의 IP Address중 동일한 IP Address는 동일한 사용자로 가정하였다. 같은 사용자가 검색 명령을 내린 이후부터 30분내에 가장 최근에 검색 명령을 내린 시각까지의 시간을 평균한 시간을 N이라 하였다. 검색 명령을 내린 후 30분내에 다시 검색을 하지 않은 검색 명령은 대상에서 제외되었으므로 N은 실제 검색 간격보다 작게 정해진다.

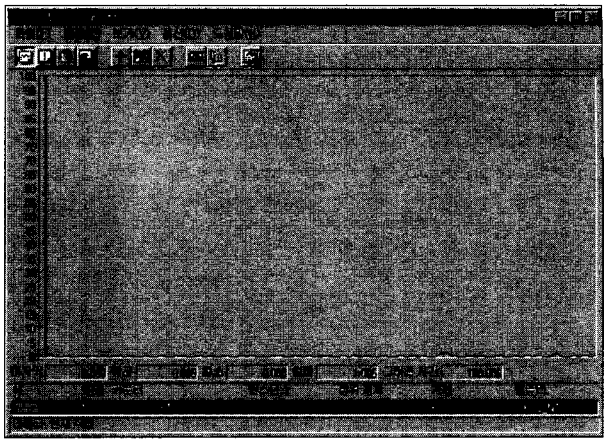
본 실험에서는 2000년 11월 10일의 로그와 2001년 1월10일의 로그를 사용하였고 구해진 시간 간격의 개수는 총 511개이며 총 시간 간격의 합은 113937초이다. 따라서 평균 검색 간격은 약 223.0초이다. 위의 정의에 따라 서버가 클라이언트로부터 223초에 M개의 검색 명령을 받으면 현재 동시 사용자 수는 M 명이라고 할 수 있다.

6.3 성능 평가 결과

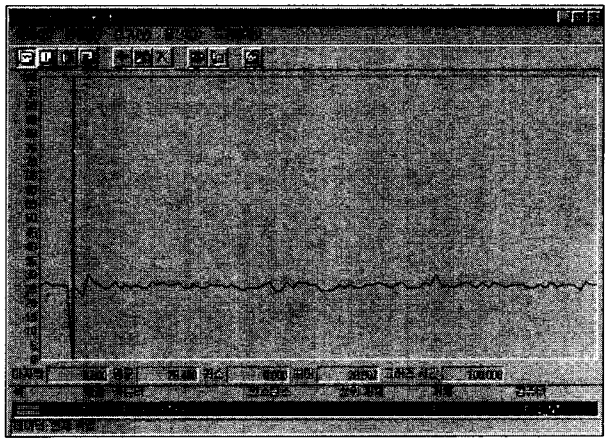
본 논문에서는 보다 효율적인 실험을 위해 동시 접속자수보다는 서버가 받는 검색 명령의 시간 간격을 기준으로 하였다. 시간 간격이 ∞, 20.27초, 2.00초, 1.00초, 0.67초, 0.50초인 경우에서 CPU 부하율을 구하였다.

(그림 11), (그림 12)는 윈도우 NT에서 제공하는 시스템 성능 모니터를 사용하여 CPU 부하율을 측정한 결과이다.

각 실험 결과에서 평균값을 구하고 해당 동시 사용자 수로 정리한 표는 <표 2>와 같다.



(그림 11) 검색 간격이 ∞일 때 CPU 부하율

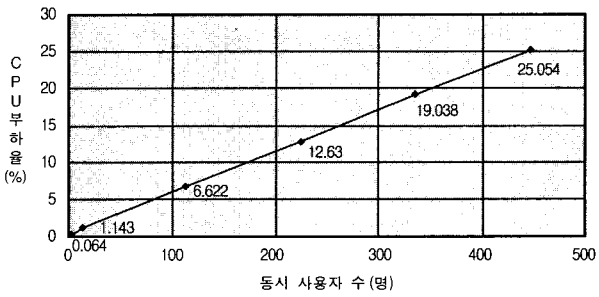


(그림 12) 검색 간격이 0.50초 일 때의 CPU 부하율

〈표 2〉 각 시간 간격에 대한 동시 사용자 수와 CPU 부하율

검색 시간 간격(초)	동시 사용자 수(명)	CPU 부하율(%)
∞	0	0.064
20.27	11	1.143
2.00	112	6.622
1.00	223	12.630
0.67	335	19.038
0.50	446	25.054

<표 2>를 그래프로 나타내면 (그림 13)과 같이 된다.



(그림 13) 동시 사용자 수에 따른 CPU 부하율

(그림 13)에서 알 수 있듯이 결과 값을 나타내는 그래프가 직선에 가까우므로 단순회귀분석을 통해 구하는 회귀직선이 실제결과와 상당히 비슷하게 됨을 알 수 있다. x를 동시 사용자 수이고 $\hat{y}$를 예상 CPU 부하율이라고 하자. 적합한 회귀직선을 구하면 다음과 같다.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{\beta}_1 = \frac{\sum x_i y_i - (\sum x_i)(\sum y_i)/n}{\sum x_i^2 - (\sum x_i)^2/n}$$

이므로 회귀직선식은

$$\hat{y} = 0.056x + 0.240$$

이 된다. CPU 부하율이 100%가 될 때의 예상 동시 사용자 수를 구하면

$$x = 1781.429$$

가 된다. 따라서 본 서버는 동시에 최대로 약 1780여명에게 서비스 할 수 있는 성능이라고 할 수 있다. 성능평가를 위한 가정에서 검색 간격인 N이 실제보다 작으므로 실제 최대 동시 사용자 수는 이보다 크다는 것을 알 수 있다.

7. 결론 및 향후 과제

본 논문은 윈도우 NT 환경에서 인터넷을 통해 서비스되는 컴퓨터 용어 사전의 구현에 대해 상세히 기술하고 성능을 평가하였다. 인터넷을 통해 사전 서비스를 제공하는 기존 사이트들의 유형과 특징을 분석하여 시스템의 구성 요소와 기능을 도출하고, 이를 바탕으로 구현하였으며 본 사전에 적합한 성능 평가 방법을 수립하여 체계적으로 성능을 분석하였다.

본 논문의 구현 내용 및 장점은 다음과 같다. 다양한 검색 방법을 지원함으로써, 사용자가 간편하고 쉽게 검색할 수 있을 뿐만 아니라 스펠링을 정확히 모를 경우에도 찾고자 하는 용어를 검색할 수 있는 기능을 제공한다. 그리고, 검색어와 관련된 HTTP사이트나 FTP사이트를 링크시켜 놓음으로써 추가 정보를 손쉽게 얻을 수 있도록 하였다. 그리고 빠르게 변화하고 새로 생겨나는 컴퓨터 용어를 신속하게 제공하기 위해 영문으로 우선 검색할 수 있게 하고 번역이 되어있는 용어는 한글로도 검색할 수 있게 하였다. 뿐만 아니라 다양한 관점에서의 의미 해석과 한국적 용어의 재 정의를 위해 한 용어에 대해서 한 명 이상의 사람이 해설을 입력할 수 있게 하였다. 또한 사용자가 용어를 검색하는 정보를 기록해둠으로써 관련용어 검색기능을 제공하고 개인별 사전이나 기타 다양한 서비스를 제공할 수 있는 여건을 만들었다.

본 논문에서는 성능 평가를 위해서 약 1년 간 운영해오고 있는 FOLDOC Mirror Site의 로그를 분석하여 사용자 검색 간격을 구하였다. 이 검색 간격을 적용하여 동시 사용자 수에 따른 CPU 부하율을 측정하였다. 이 측정 결과를 바탕으로 회귀분석을 하여 최대 동시 사용자 수를 계산해내었다. 그 결과 본 컴퓨터 용어 사전은 최대 1780여명 이상의 동시 사용자를 수용할 수 있다는 결론을 얻어내었다.

본 논문에서는 쉽게 사용할 수 있는 PC환경과 윈도우 NT를 이용하여 부가적인 소프트웨어나 하드웨어를 사용하지 않고 사용자에게 편리한 컴퓨터 용어 사전을 구현함으로써 실질적인 서비스 제공은 물론 상업적으로도 가치가 있다는 것도 보여주었다.

본 컴퓨터 용어 사전에서는 다양하고 실질적으로 도움이 되는 검색법에 대한 연구, 번역된 용어의 해설에 대한 활발한 토론을 지원할 수 있는 도구의 개발, 그리고 해설에 이미지, 사운드, 동영상을 포함시킬 수 있는 기능의 추가가 요구된다. 이 외에도 개인별 용어 사전과 같은 실용적인 통계 서비스의 개발도 필요하다.

참 고 문 헌

[1] FOLDOC, http : //foldoc.doc.ic.ac.uk/foldoc/index.html.

[2] Mosa Ma, "Agents in E-Commerce," Communications of the ACM, Vol.42, No.3, pp.79-80, 1999.

[3] 황병연, "개선된 추천을 위해 클러스터링을 이용한 협동적 필터링 에이전트 시스템의 성능", 한국정보처리학회논문지, 제7권 제5S호, pp.1599-1608, 2000.

[4] 텀즈 코리아, http : //www.terms.co.kr/.

[5] 돌도끼, http : //www.doldoki.org/dic/.

[6] 아이누리 컴퓨터 대사전, http : //dic.inuri.com/.

[7] ezWord, http : //www.ezword.co.kr/.

[8] Webopedia, http : //www.pcwebopaedia.com/.

[9] whatis?com, http : //whatis.techtarget.com/.

[10] 황병연, 박성철, "한글을 지원하는 온라인 컴퓨터 용어 사전의 개발", 한국정보과학회 춘계학술대회논문집, 제28권 제1호, 2001.

[11] 김종현, Now JavaServerPages, 한림미디어, pp.16-20, 2000.

[12] 김갑중, 황병연, "애플리케이션에서의 XML 활용경향", 한국정보과학회 데이터베이스 연구, 제16권 제2호, pp.35-44, 2000.

[13] 이종혁, "Javanuri," http : // www.javanuri.com/ jsp/user/ frm_homepage.jsp.

[14] 김세곤, "WebDox," http : //www.webdox.co.kr/lang/beans.shtml.

[15] Danny Ayers외 14명 저, 하수정 역, Professional Java Server Programming, pp.233-251, 2000.

[16] 윤영식, "JAVA DEVELOPER STUDY NETWORK," http : //www.javastudy.co.kr/docs/lec_java/jdbc/1_intro.html.

황 병 연

e-mail : byhwang@catholic.ac.kr

1986년 서울대학교 컴퓨터공학과(공학사)

1989년 한국과학기술원 전산학과(공학석사)

1994년 한국과학기술원 전산학과(공학박사)

1999년~2000년 Univ. of Minnesota Visiting Scholar

1994년-현재 가톨릭대학교 컴퓨터전자공학부 부교수

관심분야 : 공간 데이터베이스(GIS), XML, WWW 데이터베이스, 전자상거래 등

박 성 철

e-mail : schpark@postech.ac.kr

2001년 가톨릭대학교 컴퓨터공학전공 (공학사)

2002년~현재 포항공과대학교 컴퓨터공학과 석사과정

관심분야 : 전자상거래, WWW 데이터베이스, XML, Learning Machine 등