전자사전 컴포넌트의 구현

최 성 운†

요 약

사무자동화의 필요성이 증가함에 따라 많은 응용 프로그램이 개발되고 있으며, 전자사전은 이러한 사무용 프로그램의 주요 구성요소 중 하나이다. 효율적인 전자사전은 빠른 검색을 지원해야 하며, 타 사전과 데이터 교환을 통해 사어 및 신조어에 신속히 대처할 수 있어야 한다. 또한 전자 사전 프로그램 자체의 재사용을 고려하여 전자 사전 프로그램 구축비용 및 시간을 절감할 수 있어야 한다. 본 논문에서는 사전 내부 데이터 표현 형식을 정의하여 정의된 표현 방식에 기초한 타 전자 사전 데이터 교환을 가능하게 하는 방안을 제시하였다. 또한 재사용 및 호환성을 향상시키기 위하여 사전 구조를 시스템 사전 컴포넌트와 사용자 사전 컴포넌트로 나누어 구현하여 차후 바이너리 단위로의 재사용을 가능하게 하였다. 컴포넌트화로 인한 검색속도 저하 가능성은 트라이 및 B 트리 인덱스 구조를 통하여 효과적으로 방지하였다.

Component Implementation of Electronic Dictionary

Sung Woon Choi†

ABSTRACT

Many applications are being developed to automate office works, and the electronic dictionary (e-Dictionary) is one of the main components of the office suites. Several requirements are proposed for the efficient e-dictionaries : 1) Fast searching time, 2) Data compatibility with other e-dictionaries to deal with new words and obsolete words, and 3) Reusable components to develop new customized e-dictionaries with minimized development time and cost. We propose a data format with which any e-dictionary can change data with others. We also develop *System Dictionary component and Customer Dictionary component* to enable plug-and-play component reuse. Our e-dictionary achieves fast searching time by efficiently managing Trie and B-tree index structure for the dictionary components.

키워드 : 전자사전(Electronic Dictionary), 컴포넌트(Component), 재사용(Reuse), 호환성(Compatibility)

1. 서 론

인터넷의 확산과 하드웨어의 급속한 발달로 분산 시스템이 대중화되면서 기존의 데스크탑을 중심으로 한 응용 프로그램들을 인터넷을 위한 새로운 소프트웨어로 변환하고 있다. 썬 마이크로 시스템사의 StarOffice나 한글과컴퓨터사의 Netffice등은 사무용 오피스를 인터넷으로 옮긴 대표적인 예이다[1].

전자사전은 사무용 오피스의 필수 기능 중 하나로, 언어에 대한 방대한 지식을 저장 유지하는 데이터 베이스의 역할을 하였다. 이를 위해 기존 연구에서는 자연어 처리를 기반으로 검색 속도를 올리거나 적은 기억 용량을 차지하기 위한 기술에 대한 연구가 대부분이었다[1, 2]. 따라서 일반사전과는 다른 전자사전만의 사용자 편의성이나 검색 효율성에 초점을 맞추지 못하고 있으며, 또한 표준화된 사전 기술 방법이 없기 때문에 각 업체나 연구소에서 중복된 사전 개발을 수행하여 사전 데이터의 재사용 및 호환성 면에서 어려움을 갖고 있다[2, 3].

본 논문의 전자사전은 컴퓨터, 전자수첩 등의 시스템을 이용하여 쉽고 빠르게 원하는 단어를 검색할 수 있으며 사용자의 편의를 제공하는 시스템으로 정의한다. 즉, 기계 번역 정도의 단순한 작업이 아니라 사용자 인터페이스를 통하여 사용자로 하여금 아무개 문자 검색이나 기본형 검색을 지원하여 원하는 내용을 쉽고 빠르게 찾을 수 있도록 한다. 본 논문에서는 다음과 같은 요구사항을 고려한 전자 사전 개발에 목적을 둔다.

- **전자사전 프로그램의 확장성 및 재사용** : 전자 사전 프로그램에 대한 확장성 및 재사용을 위해 전자사전 소프트웨어의 컴포넌트화를 수행한다. 즉, 개발된 사전 엔진 컴포넌트(사전 COM)는 웹을 통한 서비스나 DCOM 혹은 소켓을 통한 바이너리 차원의 코드 재사용이 가능하도록 한다.
- **빠른 단어 검색** : 저장한 단어에 대한 빠른 검색은 전자

† 종신회원 : 명지대학교 컴퓨터학부 교수
논문접수 : 2001년 5월 4일, 심사완료 : 2001년 9월 20일

사전의 필수 요건이다. 컴포넌트 프로그램의 경우 표준화된 인터페이스를 만들기 위해 중간 코드인 IDL(Interface Description Language)이 필요하지만, 이 코드로 인해 일반 라이브러리를 이용한 전자사전보다 시스템 속도가 약간 떨어지는 경향이 있다. 본 논문에서 구현한 전자 사전은 시스템 사전과 사용자 사전으로 나누어 구성한다. 시스템 사전은 삽입 및 삭제가 거의 일어나지 않는 신뢰성 있는 사전 데이터를 저장한 것으로 검색 속도 향상에 중점을 두어 구현한다. 반면에 사용자 사전은 신조어 및 사어에 대처하기 위해 삽입 및 삭제가 수행되기 쉽도록 구성한다. 구분된 시스템 사전과 사용자 사전 인덱스 구조는 종합적으로 검색 속도 개선을 유도할 수 있다.

- **사용의 편이성** : 사용의 편이성은 사용자의 의도를 고려한 검색을 통해 이루어질 수 있다. 예를 들어, "showed"라는 단어가 사전 데이터 베이스에 없을 경우 show를 찾아주는 근접 매핑 기능은 사용자의 편이성을 증진시킨다.
- **데이터의 확장성 및 호환성** : 사전 데이터에 대한 확장성을 제공하기 위해 본 논문에서는 사전 데이터 형식과 문법을 정의하여 호환성을 높일 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 전자 사전 내부 데이터 표현 방식에 대한 기존 연구를 살펴보고, 본 연구의 사전 데이터 표현 방식 및 사전 정보 문법을 설계한다. 3장에서는 사전 데이터 표현 형식을 바탕으로 전자 사전 컴포넌트를 구현한다. 4장은 개발된 전자 사전의 성능을 기존의 전자사전 프로그램과 비교 분석한 실험 결과를 보인 후, 5장에서 결론 및 향후 연구방향을 제시한다.

2. 전자 사전을 위한 데이터 표현

2.1 기존 연구

전자사전 내부 데이터 표현에 대한 연구는 국내의 경우 DDMS(Dictionary Development and Management System)에서 사용한 표준 사전 표기 언어인 SDML(Standard Dictionary Markup Language)이 있다. 그러나 SDML의 경우 자연어 처리를 위한 개발 도구로 만들어졌기 때문에 발음기호, 품사정보, 파생어, 예문 등의 다양한 사전 정보를 표현하기에는 적합하지 않다[4].

국립 국어연구원에서 최근에 개발하고 있는 전자사전 시스템은 기존의 대표적인 국어 사전들을 전산화하고, 통합사전을 개발 및 관리(수정, 검색)하는 것을 목표로 하고 있다. 이 시스템은 품사, 어원, 발음, 용례, 정의, 빈도 정보 등의 사전 정보를 입력하기 위한 사용자 인터페이스 및 편집 방법 등을 정의하고 있지만 국어사전에 한정되어 있기 때문에 다국어를 지원하는 사전의 경우 적합하지 않다[5].

국외의 경우 전자북의 부각으로 북미, 일본, 유럽 등에서 활발한 연구가 진행되고 있다. 특히 일본의 전자사전연구소(EDR : Electronic Dictionary Research)에서 만든 EDR사전의 경우 일본의 통합 전자사전 시스템이며 규모가 크고 분류가 잘 되어 있다. 국어연구원의 전자사전이 국어사전에 한정된 것처럼 EDR사전 경우도 영일 기계 번역 시스템을 위해 개발되었다[6].

위와 같은 기존 연구를 토대로 전자 사전 데이터 표현이 담고 있어야 할 내용을 정리하면 다음과 같다.

1. 품사, 어원, 발음, 용례, 정의, 빈도 정보 등 다양한 사전 정보의 표현
2. 다국어 지원

2.2 전자 사전 데이터 및 문법 정의

웹 문서 표준으로는 수식이나 각종 데이터를 자유로이 표현하는 일반화된 표현 형식인 HTML이 있지만, 이는 빈도수, 발음, 품사 정보 등의 사전 고유의 특성을 지니는 형식을 표현하기에 적합하지 않다[4]. 본 시스템에서는 사전 데이터를 표현하기 위해 표제어를 기준으로 사전 정보를 담는 레코드라는 단위를 두었다. (그림 1)에서 보이는 바와 같이, 레코드에는 각 표제어의 내용이 7개의 범주로 나뉘어 표현된다.

<table>
<tr><td rowspan="2">레코드 마크</td><td colspan="4">표제부</td><td rowspan="2">변화형</td><td colspan="2">의 미</td><td rowspan="2">숙어</td><td rowspan="2">예문</td><td rowspan="2">파생어</td><td rowspan="2">참고</td></tr>
<tr><td>빈도수</td><td>표제어</td><td>발 음</td><td>품 사</td><td>스타일 정보</td></tr>
</table>

(그림 1) 사전 데이터 구조

1) 표제부 : 실제 사전을 찾는 키워드로 빈도수(초등, 중등, 고등), 표제어, 발음, 품사 정보로 구성되며 시작과 끝 태그인 <ABS> </ABS>를 가진다. 표제어는 사전내의 검색을 위한 인덱스를 제공하는 정보로 어원에 따른 낱말의 구분 번호를 지원한다.
2) 변화형 : 어원 변화, 즉, 표제어에 대한 어원 정보를 담고 있는 부분으로 <CONJ> </CONJ>의 태그 정보를 가진다.
3) 의 미 : 표제어에 대한 실제 뜻풀이를 가지고 있는 부분으로 <CONT> </CONT>의 태그 정보를 가진다. 뜻풀이 내에는 내용 풀이 외에 문법, 발음, 품사, 성구 등의 정보가 들어간다.
4) 숙 어 : 표제어에 대한 관련 숙어를 나타내는 정보를 <PHRA> </PHRA>로 구분한다.
5) 예 문 : 예문을 나타내는 태그로 <EXAM> </EXAM>으로 구분한다.
6) 파생어 : 표제어와 관련된 파생어를 넣는 부분으로 <DERIV> </DERIV>로 구분한다.
7) 참 고 : 표제어에 대한 참고 설명을 포함해서 용법이나 비슷한 말에 대한 자세한 정보를 가지고 있다. <REF> </REF>로 구분한다.

레코드 마크를 제외한 각 태그는 지칭하는 범주에서 시작

과 끝 테그를 가지며 (그림 2)의 표현 규칙을 따른다. 모든 문자 코드 체계는 유니코드를 기본으로 사용하여 다국어간 검색이 가능하도록 한다. 부록 A는 정의된 사전 데이터 표현 규칙에 따른 'test'의 사전 정보 표현을 보여준다.

```
<Dictionary>          : {<Records>}
<Records>             : '✓' <Data>
<Data>    : <Abstraction> <Contents>

<Abstraction>         : '<ABS>' [<IMP_NO>] <TEXTS>
                        [<PRON>] [<PUMSA>] '</ABS>'
<TEXTS> : <String> ['.']
<IMP_NO>              : '1' | '2' | '3'
<PRON>  : '{' <발음기호> '}'
<PUMSA>               : ' -' <품사정보> '.'
<String>  : 유니코드 스트링.
<Contents>            : [<Conjugation>] [<Content>]
                        [<Phrase>] [<Example>]
                        [<Derivation>] [<Reference>]
<Conjugation>         : '<CONJ>' '(' <CONJ_INFO>  ')'
                        '</CONJ>'
<CONJ_INFO>           : {<APHPA> [<PRON>] [<STYLES>]
                        [',' [';']]}
<Content>             : '<CONT>' {<TEXT_INFO>} '</CONT>'
<TEXT_INFO>           : {[<PUMSA>] | <CONJ_INFO> |
                        (<유니코드> [<PRON>] [<STYLES>])}
<Phrase> : '<PHRA>' {'♣' <TEXT_INFO>} '</PHRA>'
<Example>             : '<EXAM>' {'•' <TEXT_INFO>}
                        '</EXAM>'
<Derivation>          : '<DERIV>' {'㊣' <TEXT_INFO>}
                        '</DERIV>'
<Reference>           : '<REF>' {<TEXT_INFO>} '</REF>'
<STYLES>: OVERLAY | STYLE
<OVERLAY>             : ('ㅣ' | 'ㅡ' | 'ㅅ' | 'ㄴ' | 'ㄱ' | 'ㄷ' | 'ㄱ' | 'ㄴ'
                        | 'ㅁ' | 'ㅂ' | 'ㅇ' | 'ㅁ'
                        | 'ㅅ' | 'ㅁ' | 'ㅅ' | 'ㅁ' )
<STYLE> : ('ㅣ' | 'ㅡ' | 'ㅅ' | 'ㄴ' | 'ㅅ' | 'ㅁ' | 'ㅅ' | 'ㅁ')
<발음기호>            : 유니코드내의 국제 음성 부호
                        영역(U+0250 - U+02A8)
<품사정보>            : ('n' | 'ad' | 'adj' | 'v' | 'vi' | 'vt') 등의
                        품사정보
```

(그림 2) 사전 데이터 표현 규칙(EBNF 표기)

3. 웹 전자사전 컴포넌트 구현

3.1 컴포넌트 구현 기술 동향

컴포넌트란 정의된 컨텍스트와 서로 약속된 인터페이스를 통해 조합이 가능한 단위를 말한다[7]. 이러한 컴포넌트들은 ASP(Application Service Provider)나 서드파트(Third Party)들에 의해 조합해서 독립적으로 배포할 수 있다.

컴포넌트 관련 기술의 표준화는 1) 분산 기술 표준인 CORBA(Common Object Request Broker Architecture)를 기반으로 한 OMG(Object Management Group), 2) 인터넷의 자바 기술을 기반으로 한 Sun Microsystem, 3) COM(Component Object Model), OLE(Object Linking and Embedding) 등 윈도우 플랫폼을 기반으로 한 마이크로소프트의 세 조직으로 이루어지고 있다[8-10]. 마이크로소프트의 COM+는 COM과 MTS(Microsoft Transaction Server)를 통합시킨 컴포넌트 모델로 윈도우 2000과 접목하여 분산시스템에서 많은 연구 사례를 내고 있다. 이 모델은 윈도우 NT 미들웨어 컴포넌트 모델이 COM에 기반을 두고, 그 위에 MTS와 MSMQ (Microsoft Message Queue)등의 서비스를 통합한 것으로 큐 컴포넌트, 로드밸런스, 이벤트를 포함하고 있다[11-13].

본 논문에서는 COM을 기반으로 컴포넌트를 구현하여, 윈도우 NT 서버에서 웹 사전을 구축한다.

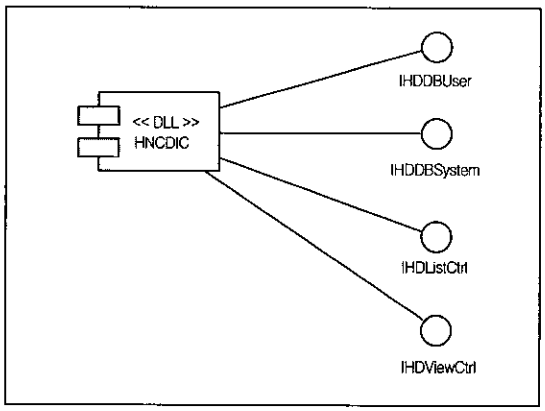
3.2 전자 사전의 컴포넌트화

전자사전을 포함한 기존의 개인용 사무 응용프로그램은 대부분 MS환경하의 윈도우즈 시스템에서 개발되고 사용되었다. 본 논문에서는 이들 프로그램에 대한 소스 코드 재이용과 개발비용을 최소화하면서 컴포넌트화를 이끌 수 있는 환경은 COM으로 판단하였다.

C++ 언어는 COM을 개발하는 언어 중에서 가장 적은 양의 소스코드를 생성하고 가장 빠른 실행 코드를 생성하는 장점이 있다. 그러나 완성도 높은 COM 객체를 만들기 위해서는 문법이나 COM 관련 지식을 습득해서 공통된 COM 인터페이스를 매번 처리해야 한다. 마이크로소프트의 컴포넌트 저작도구인 ATL(Active Template Library)는 이러한 문제를 해결하여 쉽고 빠르게 COM을 생성한다. 또한 ATL은 이용해서 생성된 컴포넌트는 목적코드가 작기 때문에 MTS 환경에서 최적의 효율을 얻는다[12, 13].

본 논문에서는 엔진에 대한 인터페이스를 시스템사전과 사용자 사전으로 나누어 구현하였다. 시스템 사전은 신뢰성 있는 사전 데이터를 저장한 사전이다. 시스템 사전은 삽입 및 삭제를 제한한 대신, 사용자 검색 속도를 높이는데 초점을 둔다. 이를 위해 다중 트라이 형태로 인덱스를 구성하였다. 또한 아무개 문자 검색이나 자소 검색 등 검색 방법을 다양화하여 사용자의 편이성을 증대하였다.

그러나 신조어에 대한 유지 보수 필요성 때문에 시스템 사전만으로는 완전한 전자 사전을 이룰 수 없다. 사용자 사전은 이에 대한 방안으로, 고어, 신조어 및 사어 등에 대처하기 위해 시스템 사전에 비해 검색 속도를 완화한 대신 삽입, 삭



(그림 3) 사전 컴포넌트 인터페이스

재 측면에서 효율성을 높였다. 이를 위해 사용자 사전은 삽입, 삭제 시 효율성 있는 B트리 형태로 인덱스를 구성하였다.

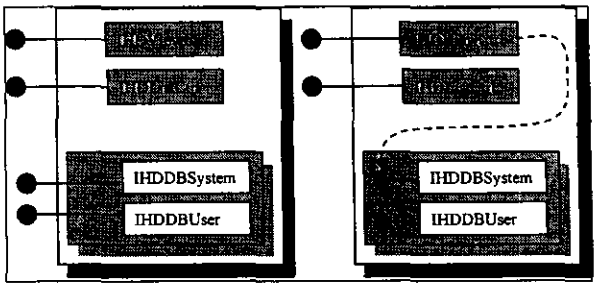
(그림 3)은 사전 엔진 컴포넌트 인터페이스를 보여준다. 시스템 사전과 사용자 사전은 IHDBSystem과 IHDDBUser 인터페이스를 통해서 사전을 검색하거나 데이터 유지보수를 한다. IHDViewCtrl과 IHDListCtrl 인터페이스는 검색 결과로 찾은 사전 데이터를 화면에 적절히 표현해 주는 인터페이스이다. 구현된 전자사전 컴포넌트는 사용자가 원할 경우 화면 출력과 관련되는 두 가지 인터페이스를 배제하고 새로 만들거나 덧붙일 수 있다. <표 1>은 사용자사전 인터페이스의 속성과 설명을 보여준다.

〈표 1〉 시스템 사전 인터페이스

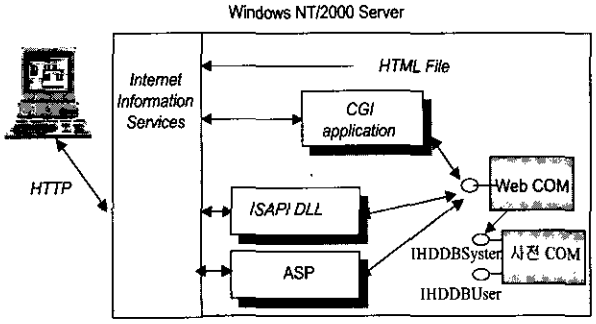
IHDDBSystem	
Property	설 명
ApproxSearch	근접 찾기를 설정/해제
InOrderSearch	순서대로 찾기를 설정/해제
FileName	사전 파일 이름을 얻어옴
DicFileOpen	사전 파일이 정상적으로 열렸는지 검사
RecordCount	사전내의 총 표제어 개수를 조사
TitleResID	사전 이름의 리소스를 가져옴
Method	설 명
Search	지정 단어 검색
Seek	레코드 포인터 이동
SetSuffixIndex	후순위 검색(Suffix Search)
SetResetIndex	선순위 검색(Prefix Search)

본 논문의 전자 사전 엔진 컴포넌트는 전자 사전 소프트웨어 구축 시 바이너리 단위의 재사용을 가능하게 한다. (그림 4-a)는 일반 데스크 탑에 탑재 가능한 전자 사전 컴포넌트를 보여준다: 이 경우 사전 엔진 컴포넌트 및 UI에 해당하는 IHDViewCtrl · IHDListCtrl을 컨테이너 객체에서 바이너리 단위로 재사용 할 수 있다. 본 시스템의 사전 컴포넌트는 비주얼 스튜디오 6.0의 컴포넌트 추가 관련 메뉴(Components and Control Gallery)를 통해서 사용자가 원하는 어떤 프로그램에서도 바이너리 코드단위로 재활용할 수 있다. 만약 웹 어플리케이션에 본 논문에서 개발한 전자 사전을 올리는 경우, 서버는 (그림 4-b)와 같은 포함(containment)형태로 작성된 컴포넌트를 재사용 할 수 있다.

(그림 5)은 본 연구에서 구현된 컴포넌트를 웹 전자사전에 이용할 경우전체 시스템 아키텍쳐를 보인다. 즉, 사전 엔진 컴포넌트인 "사전 COM"은 데스크탑이든 웹 환경이던 바이너리 차원에서 재사용된다. 그러나 실제 검색된 사전 내용을 화면에 보여주는 컴포넌트의 경우 웹은 브라우져를 통하고 데스크탑 환경은 자신의 프로그램을 통해서 보여준다. 따라서 웹의 경우는 "Web COM"이라는 HTTP서비스를 위한 컴포넌트를 따로 만들어야 한다.



(그림 4-a) Aggregation (그림 4-b) Composition



(그림 5) 웹 기반 전자 사전의 구조

4. 실 험

<표 2>는 사전 데이터 표준화를 통해서 구현된 본 논문의 웹사전 컴포넌트와 기존의 웹사전에 대해 비교 분석 한 것이다. <표 2>에서와 같이 일반적으로 사전이 가지고 있어야 할 기능에는 단순검색, 선택적 내용보기, 아무개 문자 검색, 근접단어, 데이터 컴포넌트, 데이터 추가 및 삭제, 데이터의 신뢰 등이 지적될 수 있다. 특히 선택적 내용 보기나 아무개 문자 검색 등의 기능은 사용자가 전자사전 혹은 웹사전을 찾게 유도하는 사용자 편의적 요소다.

〈표 2〉 기존 연구와의 비교

평가항목 \ 사전	본 시스템	Lycos 사전	Yahoo사전
단순검색	o	o	o
선택적 내용 보기	o	x	x
아무개 문자 검색	o	x	x
근접 단어	o	x	o
데이터 컴포넌트	o	x	x
데이터 추가/삭제	△	x	x
데이터 신뢰도	민중서림사전	?	금성출판사

아무리 사용자 편의나 기타 부수적인 기능이 많은 사전이라도 검색 속도가 느리다면 문제가 된다. 본 논문의 전자사전 검색속도를 측정하기 위해, 웹사전의 경우 클라이언트인 사용자가 서버에 사전 검색 요청을 하고 그 결과가 사용자 컴퓨터에 출력되는 시간을 기준으로 검색 속도를 측정하였

다. 실험 환경은 다음과 같다.

- 시스템 : 펜티엄 III 600EB(코퍼마인)
- 메모리 : 128M
- 인터넷 : 데이콤 보라넷 T3 전용선
- 운영체제 : Windows ME
- 벤치마크 프로그램 : WebStress 3.0

실험 대상은 현재 국내에서 사전 웹서비스를 하고 있는 라이코스와 야후를 대상으로 삼았다. 또한 객관적 평가를 하기 위해 같은 단어에 대해서 3번 측정했다. 또한 외부 요인에 의한 통신속도 변화를 감안하여 라이코스·야후·본 시스템의 순서로 test라는 단어를 검색한 후, 다시 본 시스템·야후·라이코스의 순서로 get이라는 단어를 검색하였다. 실험은 접속 상태가 양호한 새벽 3시에서 4시 사이에 실행하였다.

〈표 3〉 웹사전 검색 명령

명령 / 사전	검색명령 (test 검색 예)
Lycos 사전	http : //dic.lycos.co.kr/dic.asp?dic = ENGKOR&word = test
Yahoo 사전	http : //kr.engdic.yahoo.com/result.html?word = test
본 시스템	http : //vod.yeca.com : 8080/hncdic/dic_search.asp?keylist = off&keyword = test&auto = off

〈표 4〉 웹사전 검색 속도 비교

웹사전 / 검색단어	Lycos사전	Yahoo사전	본 시스템
test	484/506/264	66/66/66	66/76/66
get	909/736/954	154/154/142	76/66/76
a	1090/1252/1218	276/122/122	76/66/78
평균(ms)	823 ms	129 ms	71 ms

<표 3>은 실험에 사용된 웹사전 검색 명령을 보여주며, <표 4>는 상기한 실험을 하루에 1번씩 3일에 걸쳐 얻어낸 데이터를 보여준다.

<표 4>에서 알 수 있듯이, 본 논문에서 작성된 전자사전은 Lycos 사전 및 Yahoo사전에 비해 검색 속도의 현격한 향상을 보였다. 일반적으로 컴포넌트 프로그램으로 시스템을 구축할 경우 비 컴포넌트 프로그램에 비해 약간의 속도 저하는 있다. 그러나 본 시스템의 경우는 전자사전을 시스템 사전과 사용자 사전 컴포넌트로 구분하여 삽입 및 삭제가 거의 일어나지 않는 시스템 사전의 경우 검색 속도 향상을 중점적으로 이룰 수 있게 함으로써 종합적으로 속도 향상을 유도할 수 있었다. 만일 데이터에 대한 수정을 요한다면 삽입 및 삭제가 용이한 사용자 사전에서 처리하게 되어 속도면에서의 저하를 방지할 수 있었다.

5. 결 론

본 연구에서는 전자 사전의 컴포넌트화 작업을 수행하였다. 이를 위해 전자 사전 데이터에 대한 사전 내부 데이터 형식 및 문법을 정의하여 사전 데이터의 확장성을 고려하였다. 따라서 제안된 문법에 따라 사전 데이타를 표현할 경우, 서로 다른 전자사전간의 데이터 교환이 가능해진다. 또한 전자 사전 엔진의 컴포넌트를 구축하여 향후 여러 가지 종류의 사전을 출판업계에서 작성시 재사용 될 수 있도록 하였으며, 데스크탑에 탑재 가능한 전자사전 컴포넌트 및 웹 어플리케이션에서 전자사전 구현 경우 사전 엔진을 바이너리 코드 차원에서 재활용할 수 있도록 했다. 이러한 컴포넌트 작업은 신조어에 대한 유지 보수 요구가 많아지는 상황을 고려할 때 사용자의 목적에 맞는 사전 구축을 비용과 시간 면에서 절약할 수 있도록 한다. 본 논문의 전자 사전에서는 사용자의 편이를 고려한 검색을 수행하기 위해 아무개 문자 검색이나 자소 검색 등 검색 방법을 다양화하였다. IDL 중간코드로 인해 대부분의 컴포넌트 프로그램은 수행 시간이 길다는 단점이 있다. 그러나 본 시스템에서는 시스템 사전과 사용자 사전으로 구분된 사전 인덱스 구조를 트라이와 B트리 구조를 이용하여 해결하였다.

본 웹 사전에서는 신조어에 대한 판단을 하고 있지 않아 사용자로부터 들어오는 신조어의 정보가 잘못된 정보인지를 판단할 수 있는 방법이 없다. 따라서 사람을 통한 수작업을 통해 처리를 하거나 자동화된 인공지능 파서를 이용하여 사전 데이터의 신뢰성을 유지하는 작업이 요구된다. 또한 신조어를 사용자 사전에 추가할 때 정의된 사전 데이터 표현을 준수해야 하는 부담이 있다. 이를 위해 (그림 5)에서 보인 바와 같이 웹 사용자 인터페이스를 담당하는 컴포넌트(WebCom)를 두어 데이터 표현 문법과 상관없이 데이터 입력이 가능하도록 사용자의 편이성을 보장할 수 있어야 한다.

부록 A. 사전 데이터 표현 예 (test)

```
✓<ABS>3test˅1 ^[test]' -n. </ABS>
<CONT> ① ‹테스트›, 시험, 검사, 고사(achievement). [SYN.] ⇨
       TRIAL.
② 시험의 수단[방법] ; 시련 ; 시험물, 시금석.
③ '[(화학)]' 분석(시험) ; 시약 ; '[(수학)]' 검정.
④ ((영국)) ˅【야금】' 시험용 골회(骨灰)접시, 분석용 노상(爐床).
⑤ (판단·평가의) 기준 ; 시험결과, 평가.
⑥ ((구어)) = TEST MATCH.
⑦ (the T-) Ⓤ ˅【영국사】' (Test Act에 의한) 취임선서.
⑧ ˅【컴퓨터】' 시험, 테스트((제조된 논리회로의 기능·성능의
   확인)).</CONT>
<EXAM>------• a ~ in arithmetic 산수시험.①
------• a ~ for color blindness 색맹검사.
------• ⇨ ACHIEVEMENT [APTITUDE, INTELLIGENCE] TEST.
------• a ~ for carbon dioxide 이산화탄소의 검출시험.③
------• take /the Test '【영국사】 취임 선서하다.⑦
</EXAM>
<PHRA>♣'an efficiency ~ '성능검사.
♣'an oral ~ '⇨ ORAL.
♣'by all ~s '어느 점으로 보아도.
♣'give a ~ (in)' (…의) 시험을[검사를] 하다.
♣'put to the ~ '시험[음미]하다.
♣'stand '[bear, pass] 'the ~' 시험에 합격하다, 시련에 견디다.
♣'undergo a ~ '테스트를 받다. </PHRA>
```

참 고 문 헌

[1] Sun StarOffice, 한컴의 Netffice자료.

[2] 신정훈, "고성능 한글 전자 사전의 구현에 관한 연구", 대구대 대학원, 1998.

[3] 백순철, 류정준, "제한된 메모리 용량에서의 대규모 지식 처리에 관한 연구", 한국정보처리과학회, 18권 1호, pp.89-92, 1991.

[4] 사전 관리 시스템, http : //kibs.kaist.ac.kr/KLE/KIBS/Data/DDMS/ddmsA.html.

[5] 국내 사전 개발 및 관리 시스템, http : //kibs.kaist.ac.kr/beginner/tool1.htm.

[6] EDR 자연어 처리용 전자 사전 형식, http : //kibs.kaist.ac.kr/experien/etdms72.htm.

[7] 최성운, 홍선주, "CORBA 컴포넌트 모델의 분석 및 전망", 정보처리학회지, Vol.7, No.4, pp.46-52, July, 2000.

[8] Jon Siegel, "CORBA 3 Fundamentals and Programming," Joh Wiley & Sons, 2000.

[9] E. Roman, Mastering Enterprise Java Beans, John Wiley & Sons, 1999.

[10] DCOM architecture, http : //www.microsoft.com/com/wpaper/default.asp#DCOMpapers.

[11] David Cameron, "COM+ 1.0 Overview," COM+ Resource CD Online, http : //www.microsoft.com/com/resources/compluscd/overview.asp.

[12] Mary Kirtland, "Object-Oriented Sosftware Development Made Simple with COM+ Runtime Services," MJS, November, 1997.

[13] 류형규 외 3, "UML 기반 클라이언트/서버 구축", 홍릉과학출판사, 2000.

최 성 운

e-mail : choisw@mju.ac.kr

1985년 한국외국어대학교 졸업(상학학사)

1988년 미국 오레곤 주립대학교 컴퓨터공학과 (공학석사)

1992년 미국 오레곤 주립대학교 컴퓨터공학과 (공학박사)

1993년~현재 명지대학교 컴퓨터학부 부교수, OMG KSIG 회장, 한국정보컨설팅(KIC) 기술자문

관심분야 : 컴포넌트 프레임워크, 객체지향 소프트웨어공학 등