

## 멀티미디어 정보의 효율적인 검색을 위한 하이퍼미디어 시스템의 설계와 구현

正會員 高 英 坤\* 正會員 崔 潤 哲\*

### Design and Implementation of a Hypermedia System for Effective Multimedia Information Retrieval

Young Kon Ko\*, Yoon Chul Choy\* *Regular Members*

#### 要 約

하이퍼미디어 시스템은 멀티미디어 정보를 검색하고 제시하기 위하여 링크를 이용한 브라우징과 탐색 항해 도구를 이용한다. 본 연구에서 설계·구현한 시스템은 보다 효율적인 탐색항해를 지원하기 위하여 계층적 그룹과 지역 맵 도구를 제공한다. 특히, 클러스터링 기법을 적용하여 클러스터 트리를 형성하고 이를 이용하여 항해시 필요한 지식을 제공하는 방법을 제안하고 있다. 또한 탐색과 브라우징기능을 통합함으로써 하이퍼미디어 시스템의 정보검색기능과 사용자 인터페이스를 높이도록 설계하였다. 본 하이퍼미디어 시스템을 백과사전, 참고문헌정보, 전자사전, 전자책등의 분야에 적용할 수 있다.

#### ABSTRACT

Hypermedia systems have the browsing mechanism using links and provide navigation tools to retrieve and represent multimedia information. In this study we designed and implemented a hypermedia system which has the hierarchical group and local map for effective navigation. We also propose the clustering mechanism which constructs a cluster tree and uses this knowledge for navigation. The system has been designed to integrate the browsing and searching function of the hypermedia system for efficient multimedia information retrieval and user-interface. This system can be used to develop hypermedia application systems in the area of encyclopedia, reference document information, electronic dictionary and electronic book.

#### I. 서 론

정보화 사회에서 정보는 기관과 조직에서 가장 중

요한 자산으로 인식되고 있다. 이러한 정보는 그 종류가 다양하고 양적으로 매우 방대하다. 따라서 이와 같이 다양하고 방대한 각종 정보를 효율적으로 입력, 저장, 처리, 검색하고 정보간에 상호 연관성을 찾아 구조화함은 매우 중요하다. 최근 워크스테이션, CD-ROM, 광디스크(writable optical disk)등 하드웨어

\*延世大學校 電算科學科  
論文番號 : 93-124

기술의 급격한 발전과 멀티미디어(multimedia) 기술의 발전에 힘입어 텍스트뿐만 아니라 그래픽, 영상, 목소리, 사운드, 애니메이션 등의 정보를 상호 연관성을 중심으로 구조화하고, 이들을 비선형적(nonlinear)으로 접근할 수 있도록 하는 하이퍼미디어 시스템(hypermedia system)이 선진국을 중심으로 많이 개발되고 있다.

하이퍼미디어 시스템은 정보를 노드(node)와 노드들을 연결해 주는 링크(link)들의 네트워크로 구성하며, 시스템 사용자는 노드와 링크 조직을 따라 자유롭게 탐색항해(navigation)함으로써 원하는 정보를 검색할 수 있다. 따라서 하이퍼텍스트 시스템은 매우 융통성 있는 구조를 제공하고 인간의 사고(human cognition)모델과도 흡사한 장점을 제공한다. 그러나 다루는 정보 공간이 커질수록 원하는 정보를 찾는 데 많은 어려움이 따른다. Jakob Nielsen[3]은 그의 저서에서 하이퍼텍스트 시스템의 가장 큰 문제점 중의 하나로 방향 상실(disorientation)문제를 지적하고 있다. 현재까지 구현된 많은 하이퍼텍스트 시스템들은 이 문제를 해결하기 위하여 그래픽 지도(graphical map)를 제공하거나, KMS 시스템과 같이 구조적 노드(structured node)를 사용하거나[16], 혹은 노드간의 관계를 구조적으로 정의하는 등의 방식을 사용하여 왔다. 최근에는 정보공간을 좀더 효율적으로 구성함으로써 이러한 문제를 해결하려는 시도가 이루어지고 있다[10,17]. 본 연구에서는 문서 정보검색 시스템에서 사용되고 있는 검색 기법과 클러스터링 기법을 이용하여 탐색항해와 브라우징을 효율적으로 지원할 수 있는 방법을 제안하고, 이를 적용하여 하이퍼미디어 시스템을 설계·구현하였다.

## II. 하이퍼미디어 시스템

1945년 Vannevar Bush가 제안한 Memex 이후 1960년대에 들어서면서 하이퍼텍스트 시스템이 설계·구현되기 시작하였다. 초기 시스템으로는 Engelbart의 Augment/NLS, Nelson의 Xanadu, Brown 대학의 Fress 시스템을 들 수 있다[5]. 1980년대에는 대학, 연구소를 중심으로 많은 하이퍼텍스트 시스템이 개발되었는데 그 응용분야 및 목적에 따라 기능이 매우 다양하다. Conklin[5]에 따르면 응용분야에 따라 아래의 4가지로 분류해 볼 수 있다.

### 1) 매크로 문헌 시스템(macro literary system)

서로 연관되어 있는 대량의 온라인 정보를 지원하기 위한 시스템으로 초기의 NLS/Augment, Xanadu 및 1983년 개발된 TextNet을 들 수 있다.

### 2) 문제 탐색 시스템(problem exploration system)

초기에 조직되지 않은 개념들을 상호 연관시켜 주어진 문제를 해결해 나가도록 도움을 주는 시스템으로 저작 분야, 문제해결 분야, 프로그래밍 개발 분야에서 이용된다.

### 3) 구조적 브라우징 시스템(structured browsing system)

매크로 문헌 시스템보다는 적은 양의 참고 정보를 브라우징하고 검색할 수 있는 시스템으로서 사용의 용이성이 매우 중요하다. Canegie-Mellon 대학의 ZOG와 KMS, Maryland 대학의 Hyperties 및 Symbolics Document Examiner등이 여기에 속한다.

### 4) 실험적 하이퍼텍스트 시스템

몇가지 응용분야를 염두에 두고 하이퍼텍스트 기술의 실험적 응용을 위해서 개발된 시스템으로 XEROX PARC의 NoteCard, IRIS의 Intermedia, Tektronix의 Neptune등이 이 부류에 속한다.

하이퍼텍스트가 모든 정보검색 분야에 적합한 것은 아니며 Ben Shneiderman[4]은 하이퍼텍스트 시스템 분야에 적합한 응용분야를 결정하기 위하여 다음의 원칙을 판단 기준으로 제시하였다.

i) 대량의 정보가 정보단위로 구성되어 있어야 한다.

ii) 정보 단위들은 상호 연관관계가 있어야 한다.

iii) 사용자는 한 시점에서 대체적으로 매우 작은 분량의 정보만을 필요로 한다.

이러한 원칙에 입각하여 볼 때 하이퍼텍스트에 가장 적합한 응용분야로는 전자사전, 전자 백과사전, 교육, 참고문헌, 의학정보, 온라인 문헌, 소프트웨어 공학(CASE), 전자출판, 외국어 교육등을 들 수 있다. 하이퍼텍스트의 많은 장점에서도 불구하고 하이퍼텍스트의 자유로운 탐색에 기인하는 하이퍼텍스트 공간에서 사용자 방향상실이 문제점으로 부각되고 있다. 이 문제를 해결하기 위한 기법으로 전체구성도(overview diagram), 경로 추적기(path history), 안내지침(guided tour), 책갈피(bookmark), 탐색도구(searching tool), 계층구조(hierarchical structure)등이 하이퍼텍스트에 따라 선별적으로 지원되

고 있다[3]. 멀티미디어 응용분야의 확대로 지난 수년간 멀티미디어 저작도구(multimedia authoring tool)들이 개발되었다. 대표적인 저작도구로는 HyperCard, Multimedia ToolBook, IconAuthor 등을 들 수 있는데 이들은 멀티미디어 정보의 편집, 저작기능이 우수하고 제한된 범위의 브라우징 기능이 있으나 대체적으로 탐색기능이 충분치 못하고 탐색향해도구의 지원이 부족하다. 따라서 교육용 소프트웨어 제작, 전시용 멀티미디어 시스템, 안내 시스템 등의 제한된 범위내에서 매우 효과적으로 이용될 수 있다[7, 8, 11].

하이퍼텍스트 시스템의 응용분야가 매우 다양하여 하나의 일반적인 하이퍼텍스트 엔진(hypertext engine)으로는 모든 분야에 응용하기에 부적합한 측면이 있다. 본 연구에서 개발된 하이퍼미디어 시스템은 앞의 분류에 의하면 구조적 브라우징 시스템과 실험적 하이퍼텍스트 시스템 부류에 속하는 시스템으로 백과사전, 참고문헌, 전자사전, 전자책등에 활용할 목적으로 개발하였다. 특히 하이퍼텍스트 시스템의 가장 큰 문제점인 방향상실을 해결하기 위하여 브라우징과 탐색기능이 통합되도록 시스템을 설계하였다. 항해탐색(navigation)장치로 계층적 뷰, 지역 맵, 경로 추적기, 책갈피기능을 지원하였다.

### III. 정보공간(Information Space)과 정보접근방식

일상적으로 접하게 되는 정보란 정보조각들의 연속적인 묶음으로 나타난다. 이렇듯 정보를 이루는 한 단위를 정보단위(information unit)라고 볼 수 있으며, 각 정보단위를 의미적(semantic)으로 연관시켜 놓은 영역을 정보공간이라 한다[8]. 이때 각 정보단위뿐만 아니라, 정보단위 간의 연결 관계 역시 지식(knowledge)을 내포하고 있다고 볼 수 있는데, 결국 정보 검색이란 정보 공간내의 지식을 이용하여 원하는 정보단위로 접근하는 행위라고 말할 수 있다.

#### 3.1 정보공간과 모델

고전적인 하이퍼미디어 시스템의 정보공간은 정보 단위를 나타내는 노드와 각 노드간을 연결하는 링크로 구성된다. 반면에 정보검색 시스템은 주어진 질의어를 만족시키는 정보단위들의 집합을 형성하는데 필요한 지식을 정보공간의 구성요소로 가진다. 최근에는 Gloor의 CYBERMAP과 같이 노드 단위 간의 유사성을 산출하여 집합을 형성시키거나[10], 정보

단위간의 연결 관계를 이용하여 통합(aggregate)함으로써 정보공간을 구성하는 방식이 시도되고 있다[6].

Parunak은 하이퍼텍스트 시스템을 정보공간의 구성 방식에 따라 분류할 것을 제안하고 있는데, 그의 분류 방식을 따른다면 고전적인 하이퍼텍스트 시스템은 그래프식(Graph-based) 시스템으로 분류될 것이며, 정보검색 시스템은 집합식(Set-based) 시스템으로 분류될 수 있을 것이다[6].

Parunak은 또한 두 모델을 기술하기 위한 다음과 같은 간단한 형식 논리를 제안하고 있다.

(i)  $N = \{N_1..N_m\}$ 은 노드의 집합이고,  $A_{ij}$ 는 노드  $N_i$ 의  $j$ 번째 앵커(anchor)일 때  $A = \{A_{11}..A_{mn}\}$ 는 앵커의 집합이다.  $L = \langle A_{ij}, A_{k1} \rangle$ 은 링크의 집합을 나타내며, 이때 그래프식 하이퍼베이스(graph-base hyperbase)는  $H_g = \langle N, A, L \rangle$ 이다.

(ii)  $N = \{N_1..N_m\}$ 은 노드의 집합이라 하고,  $S \subseteq 2^N$ 는  $N$ 의 부분 집합의 집합일 때, 집합식 하이퍼베이스(set-based hyperbase)는  $H_s = \langle N, S \rangle$ 이다.

본 논문에서는 Parunak의 논리 형식을 확장하여 사용하였다.

#### 3.2 링크 연관(Linking Association)과 클러스터링(Clustering)

그래프식 시스템에서 정보단위 간의 의미적 연결은 링크로의 연결을 통해 이루어진다. 사용자는 링크를 통해 표상되는 노드 간의 연관 관계에 대한 지식을 이용하여 목적 노드로 접근해 갈 수 있다. 그러므로 다음과 같은 연산을 필요로 한다.

$\langle A_{ij}, A_{k1} \rangle \in L$ 이 존재하고  $N_i$ 가  $A_{ij}$ 를 가지고 있으며,  $N_k$ 가  $A_{k1}$ 을 가질 때, 노드  $N_i$ 에서 앵커  $A_{ij}$ 를 통해 노드  $N_k$ 로 도달할 수 있다. 이때  $A_{ij}$ 를 선택하기 위한 event가 변수로 작용한다.

$$f: (A_{ij}, e) \rightarrow N_k \quad (1)$$

그림1과 같은 정보공간에서 노드  $N_1$ 에서 노드  $N_5$ 로 이동할 경우 다음과 같이 나타낼 수 있다.

$$N_1 \xrightarrow{f(A_{12}, e)} N_2 \xrightarrow{f(A_{23}, e)} N_3 \xrightarrow{f(A_{32}, e)} N_4 \xrightarrow{f(A_{41}, e)} N_5$$

집합식 시스템에서 정보공간은 연관된 정보단위들

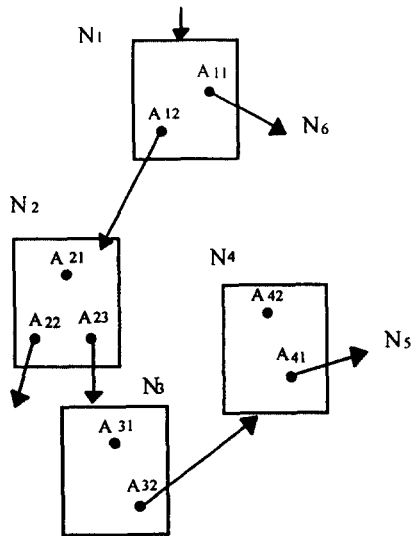


그림 1. 그래프식 시스템에서의 정보 공간

을 클러스터링함으로써 구성된다. 이때 각 정보단위는 여러 클러스터에 중복적으로 포함될 수 있다. 집합식 시스템에서의 항해는 클러스터 내의 노드를 선택하고 그 노드를 포함하는 클러스터 중 하나를 선택하는 행위를 반복함으로써 이루어진다. 이를 위해서는 다음과 같은 연산이 필요하다.

이제  $S_i \in S$ ,  $N_m \in N$ 이고,  $N_m \in S_i$ 라고 하자. 노드  $N_m$ 이 속한 집합을 알려주는 함수  $g$ 를 다음과 같이 정의하자. 이 때 한 노드는 여러 개의 집합에 동시에 속할 수 있으므로 선택을 위한 event 변수  $e$ 가 필

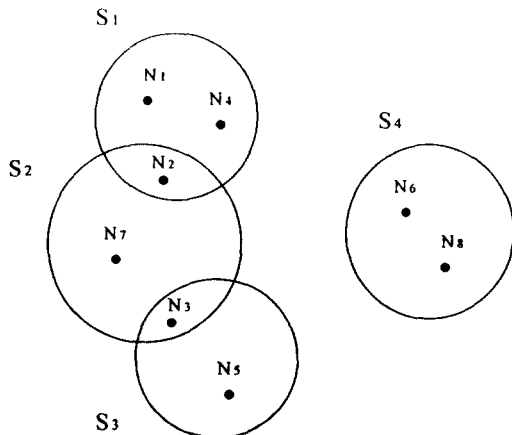


그림 2. 집합식 시스템에서의 정보공간

요하다.

$$g : (N_m, e) \rightarrow S_i, \tag{2}$$

$S_i$ 에 속한 노드 중에 하나를 선택하는 함수  $h$ 는 다음과 같다. 역시 event 변수  $e$ 가 필요하다.

$$h : (S_i, e) \rightarrow N_m \tag{3}$$

따라서 그림2의 노드  $N1$ 을 시작점으로 해서 노드  $N5$ 에 도달하는 과정은 다음과 같다.

$$N_1 \xrightarrow{g(N1, e)} S_1 \xrightarrow{h(S1, e)} N_2 \xrightarrow{g(N2, e)} S_2 \xrightarrow{h(S2, e)} N_3 \xrightarrow{g(N3, e)} S_3 \xrightarrow{h(S3, e)} N_5$$

### 3.3 통합된 정보공간에서의 항해

각 정보공간은 서로 다른 특성의 지식(knowledge)을 사용자에게 제공하기 때문에 그 나름대로의 중요한 의미를 지닌다. 그러나 서로 다른 정보공간이 가지는 의미를 분석하고 이들의 장단점을 보완하여 통합된 정보 공간을 구축하는 것은 이를 통하여 사용자가 항해시에 필요한 지식을 다양하게 제공하여 줄 수 있으므로 더욱 중요한 의미를 가진다.

Parunak의 두 모델을 단순 통합하면 다음과 같은 모델을 구성할 수 있다.

(i) 통합된 하이퍼베이스  $H_g = \langle N, A, L, S \rangle$ 로 구성된다.

(ii) (1), (2), (3)의 연산을 모두 가진다.

이 모델에서의 항해는 아래의 세 가지 방식의 혼합으로 전개된다.

(i) 연관링크(association link)를 이용하여 클러스터에 관계 없이 링크로 연관된 모든 노드에 도달할 수 있다. 노드  $N_i$ 에서 앵커  $A_{ij}$ 를 거쳐  $N_k$ 에 도달할 경우 다음과 같은 연산이 필요하다.

$$N_i \xrightarrow{f(A_{ij}, e)} N_k$$

(ii) 집합에서 노드로의 이동은 현재 노드가 속한 클러스터 내의 모든 노드가 접근가능하게 한다. 집합  $S_i$ 에서 노드  $N_i$ 에 접근할 경우 아래와 같은 연산이 사용된다.

$$Si \xrightarrow{h(Si, e)} Ni (Ni \in Si)$$

(iii) 노드에서 집합으로의 이동은 둘 이상의 클러스터에 속하는 노드를 통해 클러스터 간의 이동이 가능하다. 노드 Ni가 속한 집합 Si는 아래의 연산을 통해 얻어진다.

$$Ni \xrightarrow{g(Ni, e)} Si (Ni \in Si)$$

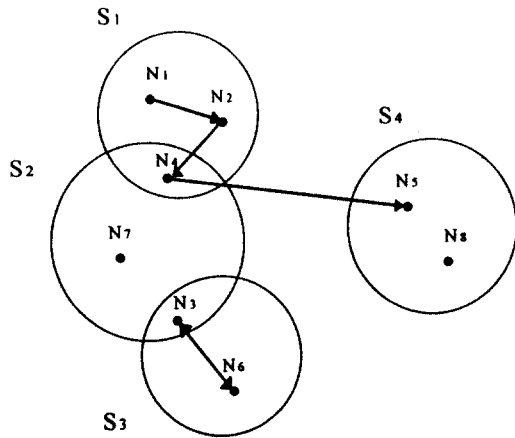


그림 3 통합된 시스템에서의 정보공간

따라서 그림3과 같은 통합된 정보 공간에서 노드 N1에서 노드 N5으로의 이동은 다음과 같다.

$$N1 \xrightarrow{g(N1, e)} S1 \xrightarrow{h(S1, e)} N4 \xrightarrow{f(A41, e)} N5$$

위에서 제시한 방법에 의하여 두 정보공간을 통합할 경우에 얻을 수 있는 효과는 적용 시스템의 특성이나 사용목적에 따라 달라질 수 있지만, 일반적인 의의를 살펴본다면 아래와 같다.

첫째, 그래프식 시스템과 집합식 시스템의 특성이 모두 보존된다.

둘째, 집합식 시스템의 한계인 다른 집합으로의 이동능력을 제공할 수 있다.

셋째, 그래프식 시스템에 자동링크의 기능을 부여하는 효과를 가진다.

넷째, 정보접근경로를 줄일 수 있다.

본 논문에서는 실질적인 시스템을 구축하여 효과적인 정보접근이 가능한지를 검토해 보고자한다.

#### IV. 하이퍼미디어 시스템의 구성과 기능

##### 4.1 정보공간의 구성

본 시스템에서의 정보공간은 그림4와 같이 노드와 링크, 클러스터 정보 및 색인정보로 구성된다.

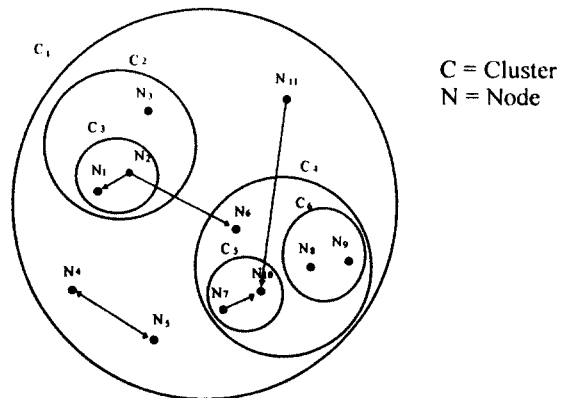


그림 4. 본 시스템에서의 정보공간

##### 4.1.1 노드의 구성

노드는 구조적으로 볼때 Collection, Subject, Item의 세가지 종류로 구분되며 각 노드는 자신에 대한 제목(Title), 요약(Description), 세부내용(Content)를 가지고 있다. Item 노드는 정보단위의 실제 내용을 가진 기본 정보단위이고, Subject 노드는 특정 주제와 밀접하게 관련된 Item들로 이루어진 하나의 개념적 노드이다. Collection은 정보를 특정 주제와 관련하여 분류화하기 위한 노드로서 모든 종류의 노드들을 구성인자로 가질 수 있다(그림6참조).

노드의 구현 방식에 따라서 텍스트 노드(Text node)와 복합 노드(Composite node)로 구분되는데 텍스트 노드는 시스템의 환경에 따라 텍스트의 구성 방식을 시스템이 융통성있게 재구성해 줄 수 있으나 복합 노드는 저자가 지정한 형식을 항상 유지한다. 이들 노드들은 모두 Item노드에 속한다. 한 노드의 내용은 여러개의 부노드(subnode)의 집합으로 이루어질 수 있다. 부노드는 독립된 노드로 존재하지는 않고 다만 Item 노드의 한 부분을 이룰뿐이다. 내용

상으로 볼때 본 시스템은 영상노드, 사운드 노드, 그래픽 노드, 텍스트 노드, 복합 노드로 나누어질 수 있는데 영상노드, 사운드 노드, 그래픽노드는 복합 노드에 속한다.

#### 4.1.2 브라우저를 위한 링크의 종류

본 시스템에서는 계층적 구조를 형성시킬 수 있도록 계층링크(Hierarchical link)를 지원하고 있으며, 각 노드의 지역 맵을 통해 연관된 노드를 탐색할 수 있도록 지역 링크(Local link)를 제공한다. 또한 본 시스템에서는 핫링크(Hot link), 참조링크(Reference link), 이동링크(Move-to link), 다중링크(Multiple link)를 지원한다(그림5참조). 핫링크는 각 단어나 절에 관한 간단한 설명을 가진 정보사전과 연결되어 있어서 단어나 절등의 설명에 사용된다. 참조링크는 하나의 노드와 연결되어 있으며 사용자가 현재 방문중인 노드를 그대로 유지시키면서 동시에 다른 노드의 내용을 살펴볼 수 있게 해준다. 이동링크는 목적노드로 이동시켜주는 기능을 가지고 있으며, 다중링크는 여러개의 목적노드를 동시에 사용자에게 제공한 후 이들의 내용을 순서대로 일련할 수 있게 해 줄뿐만 아니라 원하는 노드로 이동할 수 있는 기능을

가진다.

#### 4.1.3 클러스터 정보

각 항목 노드들의 세부 내용을 시스템이 분석하여 이 항목 노드를 대표하는 색인을 구성하고, 이를 기반으로 하여 단일링크 클러스터링(single-link clustering)방식에 따라 클러스터 트리를 형성한다. 이때 각 클러스터는 자신을 대표하는 클러스터 중심벡터(centroid)와 유사도(similarity level)를 가진다. 형성된 클러스터 트리는 사용자가 탐색할때에 특정 항목 노드와 관련된 지식을 얻고자 할 때 지식의 본체로써 사용된다. 각 클러스터의 유사도는 클러스터를 이루는 항목 노드들 간의 관련정도를 나타내므로 얻고자 하는 정보의 유사도를 지정하여 검색된 정보의 양을 자율적으로 조정할 수 있다. 시스템이 각 항목 노드를 대표하는 색인을 형성하는 과정을 좀 더 자세히 살펴 보면 다음과 같다. 우선 그 항목 노드를 이루는 모든 단어들을 검출하여 일차적인 색인 집합을 형성한다. 그 후 색인을 형성하는 집합에서 검출 빈도가 지나치게 높거나 낮은 것들은 일단 제외되는데, 이때 참조화일에 등록되어 있는 단어는 제외 대상에서 제외된다. 일차 제거후 다시 불용어 화일

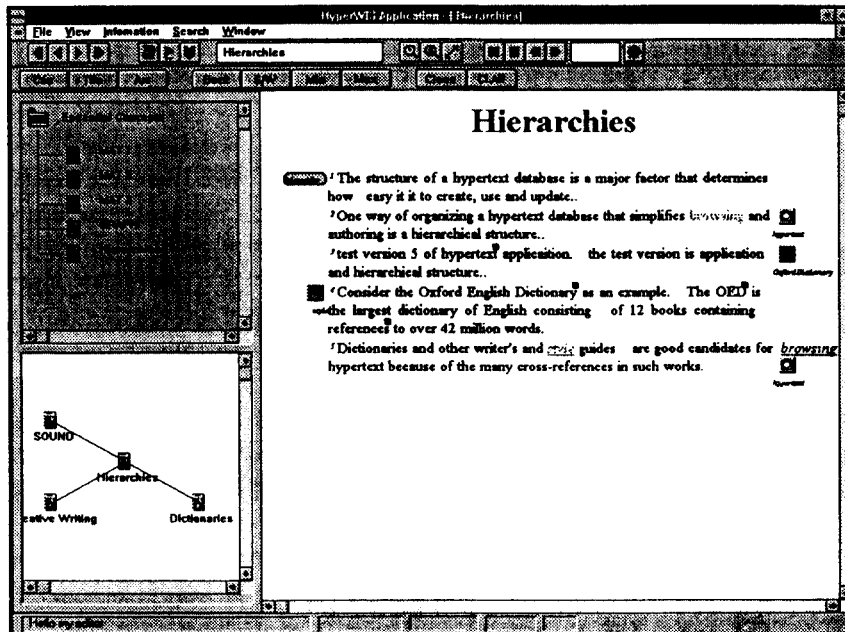


그림 5. 본 시스템의 초기화면

(stop file)을 참조하여 일반적으로 사용되는 용어들을 제거한다. 이렇게 형성된 항목 노드와 색인은 저자에 의해 재수정 과정을 거쳐 완성된다. 색인 정보는 클러스터를 형성하는데 사용될뿐만 아니라 탐색 수행 시에도 이용될 수 있다.

## 4.2 탐색항해를 위한 지식

### 4.2.1 계층적 그룹화를 통한 시작점 획득

기본 노드/링크 네트워크 구조에 근거하여 정보를 구성하는 하이퍼텍스트 시스템은 다루는 정보 공간이 커질수록 방향상실이나 인식 부담(cognitive overhead)의 문제점을 일으킨다[3]. 그러므로 사용자가 원하는 정보를 보다 쉽게 접근할 수 있는 환경을 제공하기 위하여 본 시스템에서는 기본 노드/링크 네트워크 구조 위에 계층적 그룹(Hierarchical group)을 형성한다. 이러한 계층적 그룹은 Item 노드를 Subject, Collection을 통하여 조직화하고 분류화함으로써 그림6과 같이 구성할 수 있다.

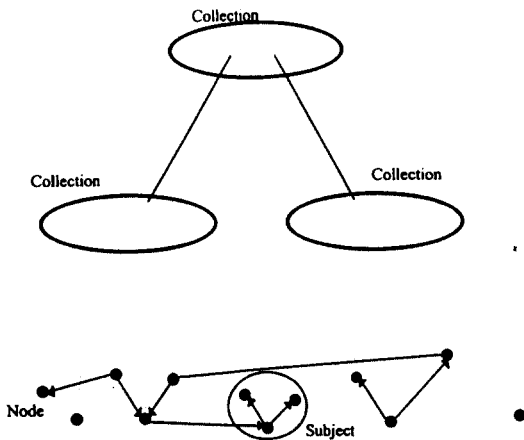


그림 6. 계층적 그룹의 구성

계층적 그룹을 구성하는데 있어 기본 개념은 추상화(abstraction) 또는 합성기능(composition mechanism)이다[18]. 합성은 관련 있는 노드들을 모아 하나의 개체로 나타내고 이 개체를 통하여 관련된 노드들에 접근할 수 있도록 하는 방식이다[19]. 이는 시스템에서 다루는 여러 노드들을 하나의 개념으로 추상화 시킴으로써 사용자가 원하는 정보 단위에 접근하는데 따른 인식의 부담을 덜어줄 수 있다. 사용자는 항해를 위한 시작 노드를 결정하기 위하여 위와

같이 구성된 계층적 그룹을 하향(top-down)방식으로 선택해 나가거나 시스템이 제공하는 탐색방법을 이용할 수 있다. 현재 이 시스템에서는 불리안방식과 확률 검색 방식을 선택적으로 사용할 수 있도록 지원하고 있다.

### 4.2.2 탐색항해를 위한 지원도구

본 시스템에서는 탐색항해를 지원하기위하여 여러 가지 정보 및 도구를 지원한다. 그중에서 탐색항해에 직접적인 영향을 미치는 것으로 계층적 뷰(Hierarchical view), 지역 맵(Local map), 클러스터 뷰(Cluster view)등이 있다(그림 5참조). 계층적 뷰는 앞에서 언급한 계층적 그룹을 실제적으로 사용자에게 보여주는 역할을 하며 시작점을 찾을때 주로 사용된다. 지역맵은 각 노드에 연관되어 있다고 판단되는 노드를 저자가 직접연결시켜 놓음으로써 사용자에게 계속적인 탐색항해를 할 수 있는 단서를 제공하는 역할을 한다. 이때 사용자가 원하는 정보를 지역 맵을 통해 충분히 얻지 못할 경우 클러스터 뷰를 사용하여 노드와 연관성이 높은 노드에 관한 정보를 얻을 수 있다. 응용 시스템을 구성할 때는 지역 맵을 이용하여 다른 클러스터 부류로 이동한 후 클러스터 뷰를 통해 관련 노드를 찾을 수 있게 구축한다면 좀 더 효과적인 탐색 항해를 지원할 수 있을 것이다.

이외에도 본 시스템은 경로 추적기나 책갈피기능을 가지고 있으며 노드의 제목을 이용해 바로 원하는 노드로 접근할 수 있는 기능도 가지고 있다. 또한 검색을 통해 원하는 노드를 얻을 수도 있는데 이에 관해서는 다음 절에서 서술하도록 하겠다.

### 4.2.3 텍스트 정보의 검색

대부분의 하이퍼미디어 시스템에서는 아주 초보적인 정보검색 기능만을 제공하는 경우가 흔한데 이는 하이퍼미디어 시스템이 기존의 정보 시스템이 가지는 한계에 대한 상반되는 개념에서 출발했기 때문일 것이다. 그러나 최근에 와서는 하이퍼미디어 시스템의 탐색항해 및 브라우징의 보조적인 수단으로 검색 기능이 새로이 주목받고 있다.

본 시스템에서는 전문색인(full-text indexing)을 통해 도치색인화일(inverted index file)을 구성하여 검색에 사용한다는 것은 앞에서도 밝힌 바 있다. 색인도 포괄적으로 본때 링크정보에 포함될 수 있을 것이다. 본 시스템에서는 구현의 편의상 링크정보와 색인정보를 따로 분류하여 처리하였다. 본 시스템이 사

용하고 있는 색인정보는 전문을 분석하여 약 200여개의 불용어를 제거한 후 도치색인화일로 저장되어 있다. 각 색인은 자신이 포함되어 있는 노드 및 부노드의 번호와 노드내에서의 물리적 위치에 관한 정보를 가지고 있어서 논리적 거리(proximity)나 물리적 거리를 이용한 검색을 가능하게 해 준다. 또한, 시소러스와 어근을 이용한 검색을 지원하고 있다.

### 4.3 시스템의 구성

그림7은 본 시스템의 전체구성도를 보여주고 있다. 그림8의 태그된 텍스트(Tagged text)는 노드/링크(Node/Link builder)를 통해 노드 및 링크정보가 추출된다. 영상 에디터(Image editor)는 실제적인 영상의 재구성 기능을 가지고 있지 않으며 다만 영상내에서 사용되는 앵커영역(Anchor region)을 구성하고 각 앵커의 계층적 관계를 형성시켜준다. 그래픽 에디터(Graphic editor)는 각각의 그래픽 객체를 형성하고 수정 및 저장할 수 있는 기능을 갖는다. 이렇게 형성된 자료는 태그된 텍스트를 통해 서로 연결된다. 색인 구성기는 태그된 텍스트에 나타나는 각각의 단어들을 추출한 후 불용어를 제거하여 도치화일로 저장한다.

위의 과정을 통하여 생성되는 데이터는 하이퍼미디어 엔진에서 사용하게 되는데 본 시스템의 엔진은 크게 노드 및 링크의 구조적관계를 관리하는 노드/링크관리자(Node/Link manager), 클러스터 정보를 관리하는 클러스터 관리자(Cluster manager), 노드의 실제적인 내용을 관리하는 내용 관리자(Content manager)로 구분되며 각각의 관리자는 통신 관리자(Communication manager)에 의해 상위 모듈과 인터페이스하게 된다.

상위 모듈 중 표현기(Visualizer)는 노드의 실제 내용을 보여 주는 기능을 담당하고 있는데 노드와 링크에 관련된 정보는 노드/링크 관리자를 통해 얻게되며 그에 따른 실제의 내용은 내용 관리자에 의해 얻어진다. 계층적 뷰는 Collection 노드의 정보를 이용하여 각 노드간의 계층적 구조를 사용자에게 보여주는 기능을 가진다. 지역맵뷰는 각노드의 내용을 보여주는 표현기에 작동되며 현 노드와 연관된 노드를 보여준다. 이때 연관된 노드란 저자에 의해 연결되어진 노드를 말하며, 클러스터 뷰에 나타나는 연관(association)된 노드는 각 노드의 내용에 따라 시스템에 의해 분류된 같은 클러스터 내의 노드들을 말한다. 이외에도 경로 관리자(History manager), 책갈

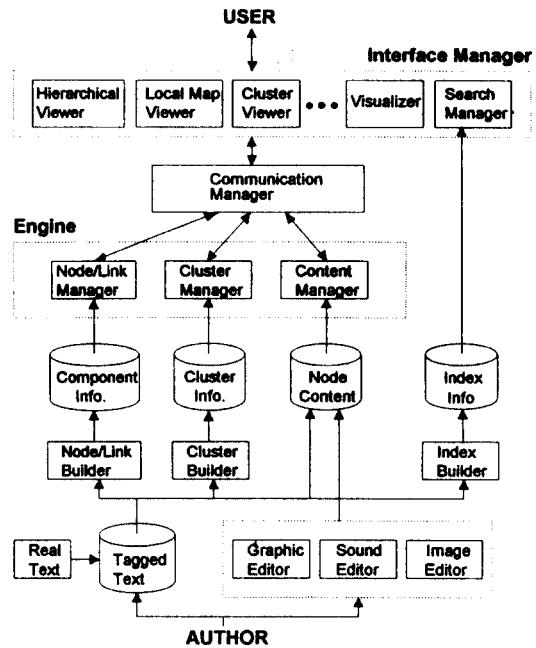


그림 7. 본 시스템의 구성도

```
(node 0015
Type linear
Title Hierarchies
text
Doc: 80 80 0 0
Title 1
( The structure of a hypertext database is a major factor that determines how
easy it is to create, use and update. )
text
Doc: 80 80 0 0
Title 2
( One way of organizing a hypertext database that simplifies <link hot in 0>-browsing-11102- and
authoring is a hierarchical structure. )
text
Doc: 80 80 0 0
Title 3
( test version 5 of <link multi in 0>-hypertext-00166-0017-0018- application.
the test version is application and hierarchical structure. )
text
Doc: 80 80 0 0
Title 4
( Consider the <link multi in 0>-Oxford English Dictionary-0019- as an example.
The <link multi in 0>-OED-0019- is the largest dictionary of English consisting
of 12 books containing <link multi in 0>-references-0019- to over 42 million words. )
text
Doc: 80 80 0 0
Title 5
( Dictionaries and other writer's and <link ref in 0>-style-0020- guides
are good candidates for hypertext because of the many cross-references in such works. )
Link
( <link ref out after 1>-browsing-0012-
<link pict out after 2>-hypertext-0016-0017-0018-
<link goto out after 3>-OxfordDictionary-001613-
<link sound out after 4>-style-0032-
<link hot out after 5>-browsing-11101-
<link pict out after 5>-hypertext-0016-0017-0018-
<link goto out before 3>-OxfordDictionary-0019-
<link sound out before 4>-style-0032-
<link ref out before 1>-browsing-0012-
<link pict out before 2>-hypertext-0016-0017-0018-
<link local out NULL 0>-NULL-0020-
<link local out NULL 0>-NULL-00168-
<link local out NULL 0>-NULL-0033-
```

그림 8. 태그된 텍스트



피 관리자(Bookmark manager)등도 상위모듈에 속하며 이들이 사용하는 정보는 통신 관리자에 의해 유지된다.

#### 4.4 자료 저장 구조

본 시스템에서 사용하는 자료로는 크게 노드/링크 정보, 노드 내용(node content), 클러스터 정보, 색인 정보 등이다. 이들 정보는 아래와 같은 구조로 저장되어 관리된다.

##### 4.4.1 노드/링크의 저장 구조

하이퍼텍스트 시스템의 구현에서 가장 중요하게 다루어야 할 부분은 정보 단위에 해당되는 노드와 그 노드를 연결하는 링크의 자료구조이다. 본 시스템에서는 Dexter 모델에 근거하여 노드와 링크의 자료구조를 형성하였다[20]. 그림9에서 보는 바와 같이 각 노드와 링크는 해당 UID를 가지며 노드는 앵커 정보에 대한 포인터를 링크는 링크 확정자(link specifier)에 대한 포인터를 가진다. 앵커 정보에는 해당 링크의 UID가 저장되어있으며 링크 확정자는 시작 노드와 목적노드의 UID를 갖는다.

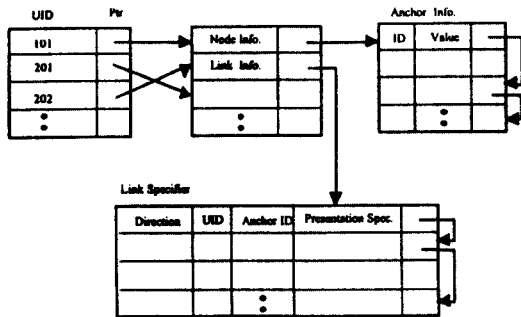


그림 9. 노드/링크 정보의 저장 구조

##### 4.4.2 노드 내용(node content)의 저장 구조

그림 10에서 보는 바와 같이 노드는 자신을 형성하는 내용을 그 성격에 따라 하나의 블록으로 묶고 이들에 관한 정보를 엔트리 파일(entry file)에 저장한다. 한페이지는 이러한 엔트리들이 모여 형성되며, 한 노드는 다시 여러 개의 페이지가 모여 형성된다. 본 시스템에서는 텍스트 노드는 하나의 페이지로 형성되지만 복합노드는 여러개의 페이지로 형성될 수 있다.

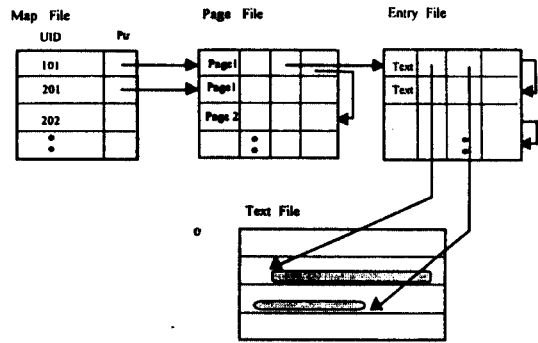


그림 10. 노드 내용 정보의 저장 구조

##### 4.4.3 클러스터 정보의 저장 구조

클러스터 정보를 유지하기 위한 자료구조는 그림 11과 같이 형성된다. 각클러스터는 해당 ID를 가지며 자신의 부모(parent) 클러스터 및 자식(child)클러스터에 대한 ID를 가지고 있다. 클러스터 트리는 개념적인 의미이며 실제로는 해쉬 테이블로 구성된다. 각각의 클러스터가 부모의 ID를 가지는 것은 이러한 구현상의 이유때문이다.

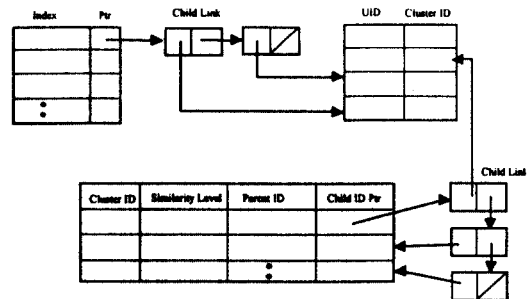


그림 11. 클러스터 정보의 저장 구조

##### 4.4.4 색인 정보의 저장 구조

색인 정보는 각 색인어 마다 자신이 속해 있는 노드의 ID, 부모노드의 ID, 노드의 시작점에서 부터의 거리(distance)를 가진다. 이때 색인어 파일의 크기를 줄이기위해 엔트리와 정보를 분리하여 저장하였다(그림 12참조).

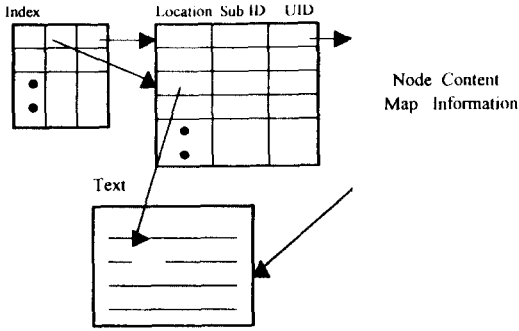


그림 12. 색인정보의 저장 구조

4.5 본 시스템에서의 정보접근 방식

그림13은 현재의 도달한 노드를 보여주고 있다. 왼쪽 하단은 관련 클러스터정보를 보여 주고 있으며 이 정보를 이용하여 원하는 노드로 이동할 수 있다. 그림15는 이동후의 모습을 보여주고 있다. 그림14에서 처럼 탐색 윈도우를 이용하여 관련 노드로 접근할 수도 있다. 그림15의 왼쪽하단에서는 클러스터 정보대신 지역맵을 이용하여 링크가 연결되어 있는 노드를 보여주고 있다. 그림16은 지역맵을 이용하여 이동한

노드의 전문을 보여주고 있다.

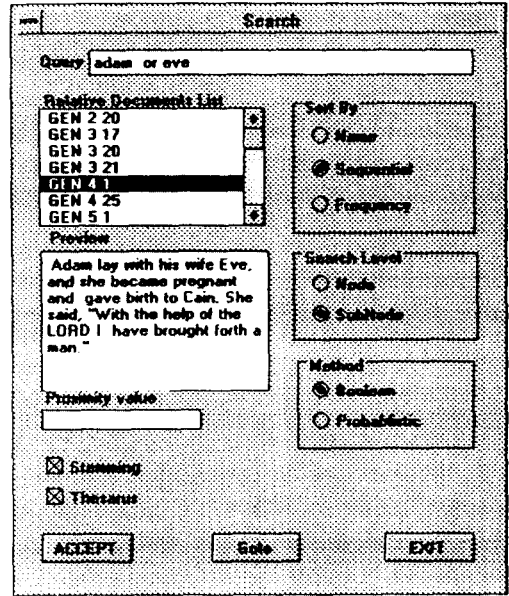


그림 14. 탐색 윈도우. 불리한 검색과 활용검색 및 proximity와 시소러스, 어근을 이용한 검색을 지원한다.

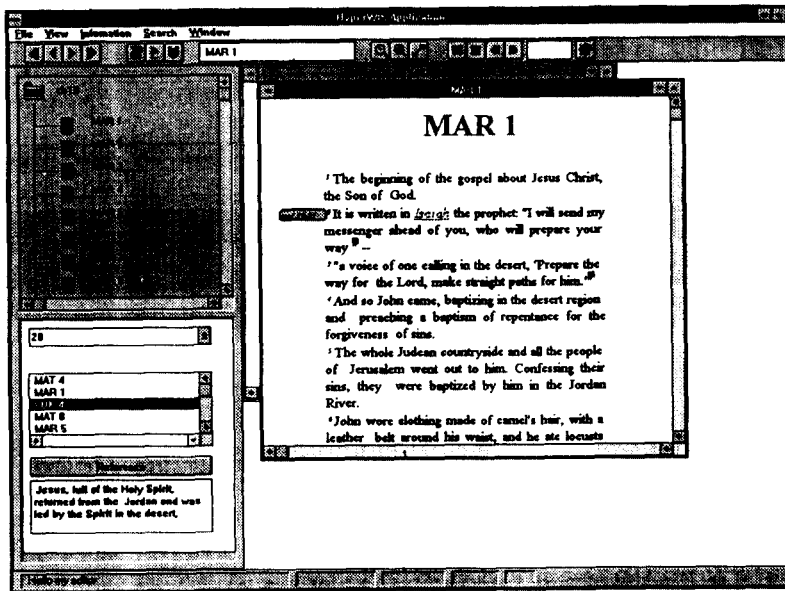


그림 13. 현재 도달한 노드. 왼쪽 아래 부분은 현재 노드와 연관성을 가진 노드들의 클러스터를 보여주고 있다.

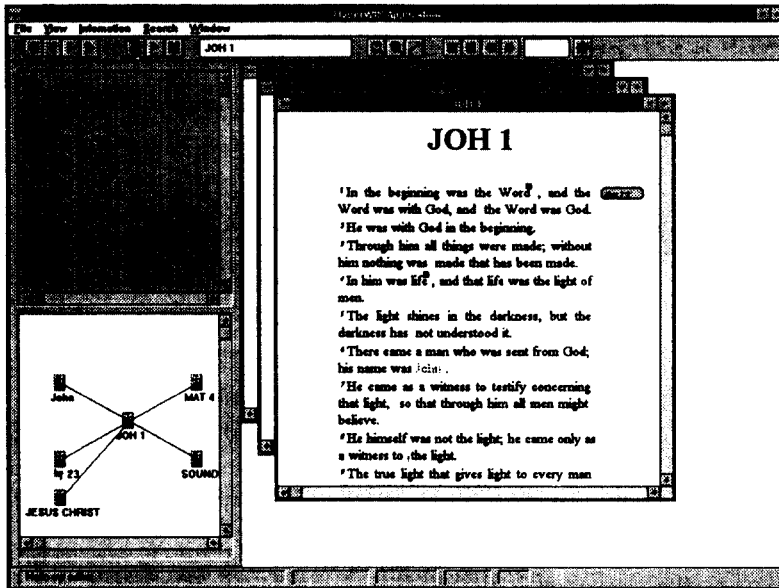


그림 15. 그림 13에서 클러스터를 이용해 도달한 노트. 왼쪽 아래 부분은 지역맵을 보여준다. 사용자는 지역맵과 클러스터 정보를 선택하여 볼 수 있다.

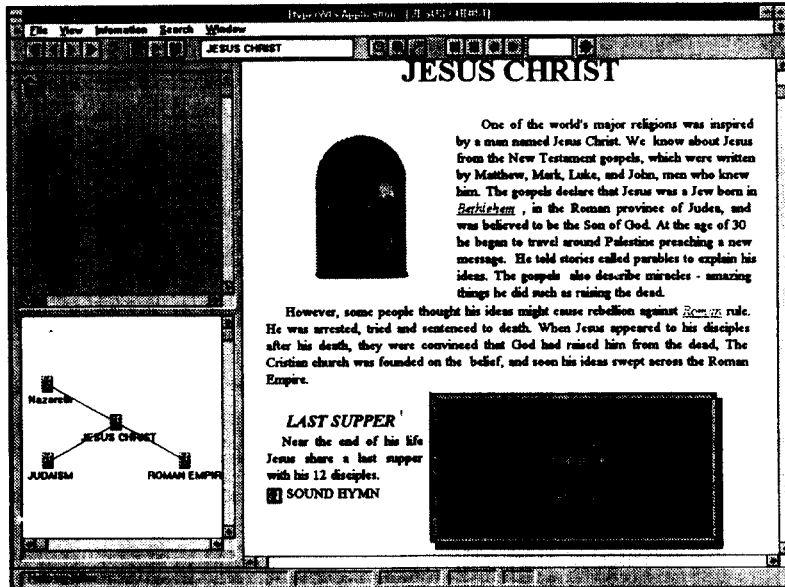


그림 16. 그림 15에서 지역맵을 이용해 도달한 노트

#### 4.6 시스템의 구현 환경

본 시스템은 IBM PC에서 개발되었으며, 운영체제로 DOS, 사용자 인터페이스는 PC Windows의 GUI를 이용하였다. 프로그래밍언어로는 MS/C++ 7.0을 사용하였다. 엔진 및 사용자 인터페이스는 C++을 이용하여 객체지향적(object-oriented)으로 설계를 하였으며, 기타 에디터 및 자료 구성기는 C를 이용하여 구현하였다. 현재 본 시스템을 이용하여 성경전문을 하이퍼텍스트화하고 있다. 본 시스템을 사용하기 위해서는 4M바이트이상의 메모리를 가진 IBM PC/386급 이상의 시스템이 필요하며, 40M 바이트 정도의 저장 용량이 필요하다. 또한 Sound Board를 갖출 경우 음성정보의 사용도 가능하다.

본 시스템을 이용하여 응용 시스템을 구축할 경우 각 응용 시스템당 본래의 태그된 텍스트보다 약 1.5배 가량의 자료 화일이 생성되며 색인의 경우는 총 텍스트 크기의 절반 가량의 색인 화일이 형성된다. 결국 한 응용 시스템을 구축하는데 필요한 자료의 크기는 원래 크기의 두배 가량이 소요된다. 현재 진행 중인 응용 시스템의 경우 태그된 텍스트의 크기는 약 9M 바이트이며, 인덱스의 크기는 5M바이트, 링크정보는 0.6M 바이트, 노드정보는 8M 바이트, 기타 정보가 4M 바이트 정도 소요되었다.

#### V. 결 론

본 논문에서는 사용자의 탐색항해를 도울 수 있는 보다 효율적인 정보공간의 구성을 위해 기존의 검색 기법 및 클러스터 정보를 결합시킨 하이퍼미디어 시스템의 설계 및 구현에 관해 살펴보았다. 본 시스템은 탐색항해시에 클러스터 정보를 이용하여 좀더 객관적인 항해 정보를 얻을 수 있다는 장점외에도 강력한 검색기능을 갖추고 있어 대량의 텍스트 정보를 하이퍼미디어화 하기에 적합할 것으로 예상된다. 현재 본 연구에서는 이 시스템을 이용하여 성경 및 백과사전을 대상으로 응용 시스템을 구축하고 있으며, 현재 성경의 경우 본문의 입력은 끝난 상태이고 각종 링크를 위한 태그작업을 진행중에 있다.

그러나 응용 시스템을 구축하는 과정에서 지역 맵과 클러스터 뷰의 상호 연계문제와 기존 화일 시스템을 사용하는데 오는 문제점 및 에디터 기능의 불충분함등이 해결해야할 점으로 나타났다. 앞으로 좀 더 강력한 저작 에디터와 독자적인 데이터베이스 시스템 구축 및 링크와 클러스터의 효율적인 연계를 위한

기능확장등의 보완 작업을 진행해 나갈 계획이다.

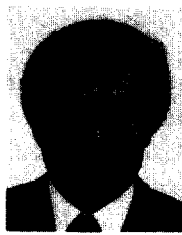
#### 참 고 문 헌

1. 신미영, "효율적인 정보 검색을 위한 클러스터 기반의 하이퍼텍스트 설계," 석사학위 논문, 연세대학교, 1992.
2. C.J.van Rijsbergen, Information Retrieval, BUTTERWORTHS, 1979.
3. J. Nielson, "Introduction to Hypertext and Hypermedia," SIGGRAPH 1990 Course Notes, No. 3, Aug. 1990.
4. Rodrigo A.B., Ben Shneiderman, "Identifying Aggregates in Hypertext Structure," Hypertext '91 Proceeding, pp.63-74, Dec.1991.
5. Conklin J., "Hypertext: An Introduction and Survey," IEEE Computer, PP 17-41, Sep. 1987.
6. H. Vann Dyke Parunak, "Don't Link Me In: Set Based Hypermedia for Taxonomic Reasoning," Hypertext '91 Proceeding, pp. 233-242, Dec. 1991.
7. Frisse M. E., "Searching for Information in a Hypertext Medical Handbook," Comm. of ACM, Vol. 31, No.7, pp.880-886, Jul.1988.
8. Mark E.F., Steve B.C., "Information Retrieval From Hypertext: Update on the Dynamic Medical Handbook Project," Hypertext '89 Proceeding, pp 199-212, Nov. 1989.
9. Parsave, K, Chignell, M, Knoshafian, s and Wong, H., Intelligent Database: Object-oriented, Deductive, Hypermedia Technologies, Wiley, 1989.
10. Peter A.Gloor, "CYBERMAP-Yet Another Way of Navigation in Hyperspace," Hypertext '91 Proceeding, pp. 107-122, Dec. 1991.
11. Walter M., "IRIS Intermedia: Pushing the Boundaries of Hypertext," The Seybold on Publishing System, August 1989.
12. Heather Brown(Ed), Hypermedia/Hypertext, CHAPMAN & HALL, 1991.
13. Poll T. Zellweger, "Scripted Documents: A Hypermedia Path Mechanism," Hypertext '89 Proceeding, pp.1-14, 1989.
14. Salton, G., Automatic Text Processing, Addi-

- son-Wesley, 1989.
15. Shneiderman B., Designing the User Interface, Secon edition, Addison-Wesley, 1992.
  16. R. Akscyn, D.L. McCracken, and E. Yoder, "KMS : A Distributed Hypermedia System for Managing Knowledge in Organization," Comm. of ACM, Vol.31, No.7, pp.820-835, July 1988.
  17. Donald B.C., Carolyn J.C. and Glenn A., "The Use of Cluster Hierarchies in Hypertext Information Retrieval," Hypertext '89 Proceedings, pp.225-236, Nov.1989.
  18. Grag P.K., "Abstraction Mechanisms in Hypertext," Comm. of ACM, Vol. 31, No.7, pp.862-870, Jul.1988.
  19. Halasz F. G., "Reflections on NoteCards : Seven Issues For The Next Generation of Hypermedia Systems," Comm. of ACM, Vol. 31, No.7, pp.836-851, Jul.1988.
  20. F. Halasz and M. Schwartz, "The Dexter Hypertext Reference Model," Proc. of the NIST Hypertext Standardization Workshop, pp.1-39, Dec.1989.



高 英 坤(Young Kon Ko) 정회원  
1967년 2월 4일생  
1991년 : 한국과학기술대학 전산학과 학사  
1993년 : 연세대학교 전산학과 석사  
1993년~현재 : 연세대학교 전산학과 박사과정



崔 潤 哲(Yoon Chul Choy) 정회원  
1950年 6月 20日生  
1973년 2월 : 서울대학교 전자공학과 졸업  
1975년 6월 : University of Pittsburgh(공학석사)  
1979년 6월 : University of California, Berkeley, Dept. of IE & OR(공학박사)  
1979년 8월~1982년 7월 : Lockheed사 및 Rockwell International사 책임연구원  
1982년 9월~1984년 1월 : University of Washington, 전산학과 박사과정  
1990년 9월~1992년 1월 : University of Maisachusetts, 연구교수  
1984년 3월~현재 : 연세대학교 전산학과 부교수  
※주관심분야 : 멀티미디어 및 하이퍼미디어, 지리정보시스템(GIS), 전자사전등