■ 應用論文

Journal of the Ergonomics Society of Korea
Vol 12, No.1 1993

효율적인 탐색과 브라우징을 지원하는 하이퍼미디어 시스템의 사용자 인터페이스 설계
(User-interface Design of a Hypermedia System for Effective Searching and Browsing)

고영곤* · 최윤철*

Abstract

Hypermedia systems allow the retrieval and representation of multimedia information such as text, graphics, image and voice/sound using navigation and browsing mechanisms. In this study we developed a hypermedia system which provides hierarchical group, local map and cluster view for effective navigation in hyperspace. The system also supports hot link, reference link, move-to link and multiple link to browse the multimedia information space effectively. This system has been designed to integrate the navigation, browsing and searching function of the hypermedia system in human factor perspective and provides the user-friendly user interface mechanism.

Ⅰ. 서 론

정보검색 시스템과 같은 종래의 정보 시스템이 가지는 한계는 이를 사용하는 사용자에게 이미 그 시스템에 대한 정확한 지식과 자신이 찾고자하는 정보의 성격에 대한 이해 및 이를 효율적으로 표현할 수 있는 능력을 요구한다는 점이다. 또한 최근 급속한 컴퓨터 기술의 발전으로 텍스트 정보뿐만 아니라 영상, 그래픽, 사운드등의 멀티미디어 정보에 대한 처리가 요구되고 있는데 기존의 데이타베이스 시스템은 이러한 다양한 정보를 처리하기에는 미흡하다. 이러한 배경에서 기존 정보시스템의 한계를 극복하고 사용자 입장에서 좀더 편리하고 쉽게 원하는 정보에 접근할 수 있는 방법에 대한 연구가 진행되어 왔다. 이에 따라 최근 연관적인(associative) 인간의 사고(human congnitive)과정과 유사하게 정보를 비순차적(nonlinear)으로 접근할 수 있는 하이퍼미디어 시스템(hypermadia system)에 대한 연구와 개발이 많이 수행되어왔다[3]. 지금까지 개발된 Brown 대학의 Intermedia, Maryland 대학의 Hyperties나 Apple사의 HyperCard 및 Guide와 같은

*연세대학교 전산과학과

하이퍼미디어 시스템들은 이러한 측면에서 상당한 성공을 거두고 있으며, 이외에도 XEROX PARC의 NoteCards나 Textronix의 Neptune과 실험용 시스템들 역시 하이퍼미디어 시스템의 가능성을 보여주고 있다[5].

그러나 하이퍼미디어 시스템이 가지는 이러한 장점에도 불구하고 하이퍼미디어 시스템의 설계와 응용은 몇가지 문제점을 내포하고 있는데, 그 중에서도 특히 정보공간에서의 방향상실(disorientation)이 가장 큰 문제점으로 부각되고 있다. 따라서 관계형 데이타베이스 시스템이나 정보검색 시스템의 문제점을 메꿀수 있는 새로운 형태의 하이퍼미디어 시스템을 개발하려는 노력이 진행되고 있으며, 본 논문은 이러한 시도의 일환으로 기존의 정보 검색 기법을 보조적인 수단으로한 탐색항해(navigation)를 통해 보다 효과적으로 정보에 접근할 수 있는 하이퍼미디어 시스템을 구축하고자 한다.

Ⅱ. 하이퍼미디어 시스템

1945년 Vannevar Bush는 "As we May Think"라는 논문을 통해 종래의 정보탐색 방식은 각각의 정보에 접근하는 경로가 오직 하나뿐임을 지적하고, 이는 인간이 스스로 많은 경로를 통해 정보를 획득하고 유지한다는 것에 상반된다고 비판하였다. 그는 인간의 정보 접근방식처럼 여러 경로를 통해 정보에 접근할 수 있도록, 한 정보단위에서 다음 관련된 정보단위로 링크를 연결하는 연관적 색인(associative indexing)이라는 개념을 제안하였다. 이후 Engelbart의 Augment/NLS, Carnegie-Mellon 대학의 KMS, Brown대학의 Fress등이 개발되면서 하이퍼미디어의 가능성을 보여준 이후 꾸준한 발전을 거듭하여 현재 Intermedia, Guide, HyperCard, Hyperties등이 상용화되어 널리 사용되고 있으며, 새로운 개념의 하이퍼미디어 시스템들이 계속 개발되고 있다[5]. 그러나 하이퍼텍스트가 모든 정보검색 분야에 적합한 것은 아니며 Ben Shneiderman[4]은 하이퍼텍스트 시스템 분야에 적합한 응용분야를 결정하기 위하여 다음의 원칙을 판단 기준으로 제시하였다. 첫째, 대량의 정보가 다수의 정보단위로 구성되어야 하고, 둘째, 정보단위들은 상호연관 관계가 있어야하며, 셋째, 사용자는 한 시점에서 대체적으로 매우 작은 분량의 정보만을 필요로 해야한다. 이런한 원칙에 입각하여 볼때 하이퍼미디어 시스템의 가장 적절한 응용분야로는 전자사전, 전자 백과사전, 교육 분야, 참고문헌 분야, 의학정보 분야, 온라인 문헌 분야, 소프트웨어 공학(CASE), 외국어 교육등을 들 수 있을 것이다[7, 8, 11].

그림 1에서 보는 바와 같이 일반적으로 하이퍼미디어 시스템에서 각 정보단위는 노드(node)로 표현되며, 이들 노드들은 링크(link)를 통해 상호 연결되어 있어서[3], 사용자가 여러가지 경로를 통해 노드에 접근할 수 있게 해준다. 또한 사용자의 정보접근을 지원하기 위해 여러가지 항해탐색 도구를 지원하고 있는데 책갈피(bookmark), 경로 추적기(path history), 전역맵(global map) 등이 이에 속한다. 이러한 지원도구들 외에도 질의어를 통해 자신이 접근하고자 하는 노드를 기술할 수 있도록 하

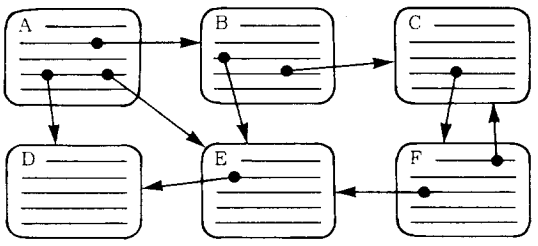

그림 1. 하이퍼텍스트에서 노드와 링크의 관계

거나 관련 노드를 동적(dynamic)으로 연결하는 등의 시도가 이루어지고 있으며, 한편으로는 정보검색 기법과 인공지능 기법의 접목도 시도되고 있다.

하이퍼미디어 시스템과 관련된 연구 동향은 크게 세가지로 나누어 볼 수 있다. 한 가지는 기존의 하이퍼미디어 시스템의 문제점을 보완하거나 해결하기 위해 새로운 개념의 브라우징 및 탐색기법을 고안하고 이를 지원하기위한 시스템을 개발하려는 시도이며, 다른 하나는 현장연구(field study)등을 통해 인간적 요소(human factor)를 고려하여 사용자가 좀 더 자연스럽게 시스템에 접근할 수 있도록 화면의 구성 양식이나, 노드의 크기, 링크의 표현 방식, 입력 장치등의 효과적인 설계를 시도하는 것이다. 또 다른 연구방향은 기존의 하이퍼미디어 시스템의 기능을 이용하여 효과적인 응용 시스템을 구축할 수 있는 방법을 연구하고 이를 다른 응용 시스템의 설계에 적용하는 것이다.

그러나 이러한 몇가지 연구방향은 결국 사용자가 자신이 원하는 정보를 가장 편리하고 효율적으로 획득할 수 있도록 하고자 한다는 면에서 같은 맥락에 속한다고 볼 수 있을 것이다. 즉 사용자와 컴퓨터간의 인터페이스를 좀 더 자연스럽게 연결하고자 하는 의도에서 비롯된 것으로 이해할 수 있을 것이다. 어떤 시스템에 대한 판단은 그 기능보다는 오히려 사용환경에 더 좌우된다는 인식이 지난 10여년전부터 계속되어오고 있으며[12], 이러한 사실은 대부분의 시스템 설계시에 크게 반영되고 있다는 것을 고려할때 이러한 연구추세는 앞으로 계속되리라 예상된다.

본 연구에서는 대량의 텍스트 정보를 포함한 멀티미디어정보를 다룰 수 있는 하이퍼미디어 시스템을 구축하였는데 특히 백과사전, 참고문헌 정보, 전자사전(electronic dictionary), 전자책(electronic book)등에 활용할 목적으로 개발하였다. 본 시스템은 효율적인 이용환경을 구축하기 위하여 여러가지 탐색 및 브라우징 도구를 지원하고 있다. 특히 하이퍼미디어 시스템의 가장 큰 문제점인 방향상실을 해결하기 위하여 브라우징(browsing)과 탐색(searching) 기능이 통합되도록 시스템을 설계하였다. 또한 탐색항해를 지원하기 위해 계층적 뷰(hierarchical view), 지역 맵(local map), 경로추적기(path history), 책갈피(bookmark) 기능등을 가지고 있다. 이들을 통해 기존의 검색 시스템의 장점을 취하는 동시에 좀 더 편리한 사용자 인터페이스를 제공할 수 있는 방법을 제시하였다.

Ⅲ. 정보탐색과 브라우징

3-1. 정보탐색

일상적으로 접하게 되는 정보란 정보 조각들의 연속적인 묶음으로 나타난다. 이렇듯 정보를 이루는 한 단위를 정보단위(information unit)라고 볼 수 있으며, 각 정보단위를 의미적으로 연관시켜 놓은 영역을 정보공간(information space)이라 한다[8]. 이 때 각 정보단위뿐만 아니라 정보단위간의 연결 관계 역시 지식(knowledge)을 내포하고 있다고 볼 수 있는데, 결국 정보탐색이란 정보공간내의 지식을 이용하여 원하는 정보단위로 접근하는 행위라고 볼 수 있다.

3-2. 정보공간과 모델

고전적인 하이퍼텍스트 시스템의 정보공간은 정보단위를 나타내는 노드와 각 노드간을 연결하는 링크로 구성된다. Parunak[6]은 하이퍼텍스트 시스템을 정보공간의 구성 방식에 따라 분류 할 것을 제안하고 있는데, 그의 분류방식에 따른다면 고전적인 하이퍼텍스트 시스템은 그래프식(graph

-based)시스템으로 분류될 수 있다. 그래프식 시스템에서 정보단위 간의 의미적 연결은 링크로의 연결을 통해 이루어진다. 사용자는 링크를 통해 표상되는 노드간의 연관관계에 대한 지식을 이용하여 목적노드로 접근해 갈 수 있다. 즉, 그림 2에서와 같이 노드 N1에서 노드 N5로 이동할 경우 사용자는 노드 N1의 앵커 A12를 이용하여 노드 N2로 이동한 후 다시 앵커 A23를 이용하여 노드 N4로 이동한 후 다시 노드 N5에 도달할 수 있게 된다.

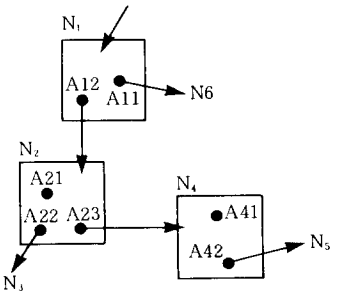

그림 2. 그래프식 시스템에서의 정보공간

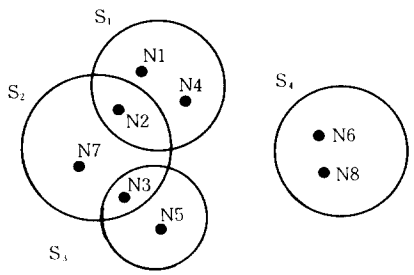

그림 3. 집합식 시스템에서의 정보공간

반면에 기존의 검색 시스템은 집합식(set-based)시스템에 가까운데, 이러한 집합식 시스템에서 정보공간은 연관된 정보단위들을 클러스터(cluster)로 묶음으로써 구성된다[6]. 이때 각 정보단위는 여러 클러스터에 중복적으로 포함된다. 이러한 시스템에서의 탐색행해는 그림 3에서 보듯이 클러스터 내의 노드를 선택하고 그 노드를 포함하는 클러스터 중 다른 하나를 선택하는 행위를 반복함으로써 이루어진다.

결국 그래프식 시스템에서는 연관된 정보에의 접근이 노드간의 연결관계에 지나치게 의존하고 있어서 오히려 연관 정보로의 접근을 방해할 소지가 있는 반면, 집합식 시스템에서는 서로 다른 부류로의 이동이 쉽지 않은 단점을 가진다. 그러나 각 정보공간 구성방식은 나름대로의장점을 가지고 있기 때문에 이들 정보공간은 보완적으로 통합할 경우 연관성이 높은 노드로의 이동 뿐만 아니라 다른 부류의 클러스터 집합으로의 이동도 용이한 정보공간이 형성될 수 있다.

3-3. 통합된 정보공간에서의 탐색항해

그림 4는 두개의 정보공간을 단순 통합하여 구성한 정보공간을 보여주고 있다. 이때 통합된 정보공간에서의 노드 N1과 N3는 비록 링크가 연결되어 있지 않으나 N1과 N3가 근접한 클러스터 부류에 속하기 때문에 쉽게 이동이 이루어질 수 있다. 반면 N2와 N6는 서로 다른 클러스터 부류이지만 링크를 이용해 쉽게 이동할 수 있다. 즉 두 정보공간의 통합은 가까운 거리의 정보들은 집합을 통해 파악할 수 있으며 링크는

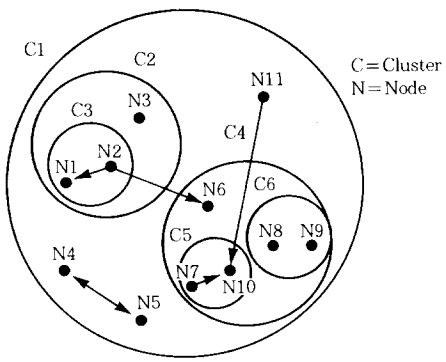

그림 4. 통합된 모델에서의 정보공간

다른 부류의 정보군으로 이동할 때 쓰일 수 있다는 장점을 가진다. 본 연구에서는 이러한 개념에 입각하여 하이퍼텍스트의 브라우징 방식과 클러스터링 방식을 통합함으로써 보다 효율적인 탐색항해가 가능하도록 설계하였다. 즉 현 시스템에서 근거리의 관련 노드는 클러스터를 통하여 획득할 수 있으며 원거리의 클러스터 부류로의 이동은 링크를 통해 이루어질 수 있다.

Ⅳ. 하이퍼미디어 시스템의 구성과 기능

4-1. 멀티미디어 노드의 구성

노드(node)는 정보의 기본 단위로 사용된다. 그러므로 정보의 종류가 다양하듯이 사용 노드의 종류도 다양하다. Hyperties와 같은 경우에는 오직 한가지 종류의 노드만을 사용하고 있는데 이 경우에는 사용자에게 일관성을 제공해 줄 수는 있으나, 멀티미디어 정보를 표현하기에는 어려움이 따른다. 본 시스템에서는 여러가지 종류의 노드들을 사용하고 있는데 우선 구조적인 측면에서 Collection 노드, Subject 노드, Item 노드로 나누어진다(그림 5 참조). 일반적으로 사용자들은 특정 노드에 접근할 때 계층적으로 접근하는 것에 익숙해 있다.가령 성경과 같은 경우 특정 구절을 찾고자 할 때 우선 책(Book)을 찾고, 다시 장(Chapter)을 찾은 후 절을 찾게된다. 이러한 계층적 접근을 지원하기 위하여 사용자가 접하게 될 실제적인 정보를 가지고 있는 Item노드란 것을 두었으며, 계층정보를 가진 Collection노드를 그위에 구성한다. Collection노드는 관계된 특성을 가진 Item 노드들을 한 부류로 인식할 수 있도록 묶어 주는 역할을 한다. 또한 Collection 노드내에 다른 Collection 노드를 내포할 수 있으며 서로 다른 Collection이라도 같은 Item노드를 가질 수 잇다. 다만 순환적 구조(cyclic structure)가 형성되는 것을 방지하기 위해 여러 개의 Collection이 같은 Collection을 포함할 수 없도록 하였다. Subject 노드는 Item 노드와 Collection 노드의 성격을 조금씩 취하고 있다. 즉 여러 개의 Item 노드를 하나로 묶어서 대표하는 성격을 띄지만 Collection 노드처럼 Item 노드를 내포하는 것은 아니며 단지 각 Item 노드들이 어떤 관계로 묶어져 있는지를 보여주는 대표자의 성격을 가진다.

본 시스템의 노드는 위와 같은 구조적인 측면외에도 구현 측면에서 나누어 볼 수도 있다. 노드의 구현 방식에 따라서 텍스트 노드(text node)와 복합 노드(composite node)로 구분되는데 텍스트 노드는 시스템의 환경에 따라 텍스트의 구성방식을 시스템이 융통성있게 재구성해 줄 수 있으나 복합노드는 저자(author)가 지정한 형식(format)을 항상 유지한다. 이와 같이 함으로써 텍스트 정보를 표현할 윈도우의 크기나 위치 및 상태에 따라 적절하게 시스템이 형식을 재구성할 수 있다. 성경이나 일반적인 책의 경우 각 페이지의 형식(format)을 일일이 파악하여 지정해준다는 것은 비효율적일 뿐만 아니라 시스템의 융통성을 저해할 가능성이 크다. 또한 사용자 입장에서 볼 때에도 시스템의 환경이 변함에 따라 텍스트의 형식도 따라 변하는 것이 자연스럽게 느껴질 수 있다. 복합 노드는 백과사전과 같이 한 페이지를 구성하는 형식이 정형화되어있지 않을 경우에 사용되는 것으로 각 텍스트 정보나 그래픽, 이미지 정보가 나타날 위치, 크기등을 미리 정해둠으로서 사용자가 시스템의 사용환경을 변화시키더라도 정보의 구성 형식에는 변화가 없다. 이러한 노드는 지도(map)나 브라우징 맵(browsing map)을 구성하는데 사용되기도 한다. 이외에도 이미지 노드, 그래픽 노드등이 있는 데 이들은 모두 복합 노드에 속한다.

또한 한 Item 노드에는 여러개의 부노드

(subnode)가 존재할 수 있는데 이들은 탐색 및 정보 검색시에 목적앵커(destination anchor)로 사용될 수 있다. 노드를 지나치게 세분화 하는 것은 때로는 사용자에게 혼란을 가중 시킬 수 있는데 가령 성경의 각 구절을 노드화 한다거나 백과사전에서 한 엔트리의 설명을 여러개의 노드로 나누는 경우이다. 본 시스템은 지나친 세분화를 방지하면서도 사용자가 한 노드의 특정한 구절이나 문단으로 바로 이동할 수 있게 하기 위해서 부노드의 개념을 도입하였다.

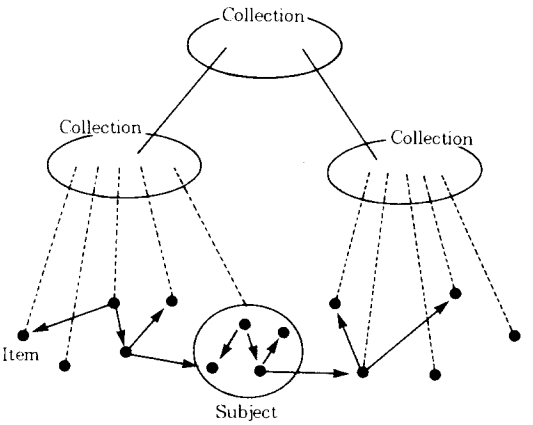

그림 5. 계층적 그룹의 구성

4-2 브라우징을 위한 링크의 종류

Bush는 특정한 노드에 도달하는 여러가지 경로를 제시하기 위해 링크의 연속적인 연결인 트레일(trail)개념을 제시하였다[13]. 이러한 링크 트레일의 개념은 인간이 상황에 따라 여러가지 경로를 따라 같은 정보에 접근하는 것과 유사한 방식으로 인식되고 있으며 많은 하이퍼미디어 시스템에서 이를 지원해 주고 있다.

본 시스템에서 사용되고 있는 링크(link)들은 모두 단방향 링크들로서, 목적노드(destination node)의 종류 및 연관 정도에 따라 다른 링크를 사용한다. 하이퍼미디어 시스템을 사용하는 사용자는 본문 중에 나오는 단어나 지명, 인명등에 대해 간략한 설명을 원하는 경우가 있는데 이때 그 단어나 구절에 관계된 절로 완전 이동하는 것은 사용자로 하여금 방향상실을 초래할 우려가 있다. 이를 방지하기 위하여 본 시스템은 사전을 이용하여 각 사전의 엔트리(entry)와 특정 단어 및 구절을 링크로써 연결하고 있다. 사전은 연결된 단어 및 구절에 대한 간략한 설명을 가지고 있으며 같은 단어일지라도 여러개의 엔트리를 가질 수 있게 구성 되어있다. 이렇듯 사전과 연결되어 있는 링크를 본 시스템에서는 핫링크(hot link)라고 부르는데 이 핫링크의 앵커영역(anchor region)은 그 중요성을 지나치게 부각시키지 않기 위해서 본문의 글자꼴(font)과 같은 글자꼴을 사용하며 다만 약간 더 밝은 색을 띄고 있다. 사용자가 이 링크를 선택할 경우 간단한 메세지 윈도우(message window)를 통해 설명이 나타나며 사용자에 의해 쉽게 제거 될 수 있다.

만약 사용자가 좀 더 자세한 설명을 원할 경우 핫링크로는 그 기능을 충분히 수행할

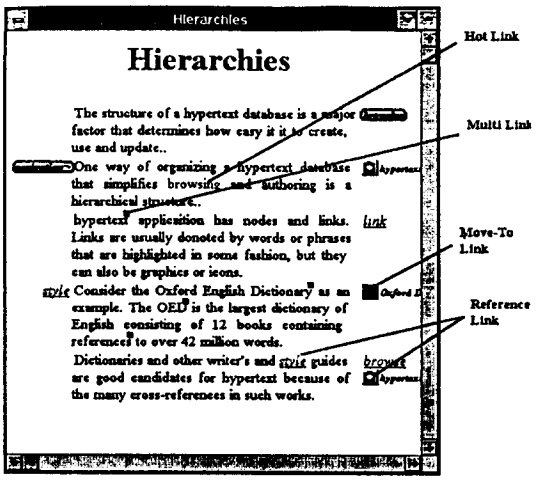

그림 6. 본 시스템이 지원하는 링크의 종류

수 없다. 왜냐하면 핫링크의 목적앵커는 또 다른 링크를 가질수 없으며 아미지나 그래픽, 사운드 정보 역시 가질 수 없기 때문이다. 이에 따라 참조 링크(reference link)라는 것을 두었는데 참조링크의 목적앵커는 Item 노드 및 Item 노드의 부노드가 될 수 있다 .이 참조링크는 이탤릭체에 밑줄을 가진 형태로 나타나서 핫링크 보다는 중요도가 좀 더 부각될 수 있다. 참조링크에 의해 참조되는 노드는 새로운 윈도우를 통해 나타나며 시작노드와 함께 계속적으로 유지될 수 있다. 반면 이동링크(move-to link)는 시작노드가 사라지고 목적노드가 현재 윈도우(current window)에 나타나게 되어있다. 이동링크의 목적노드는 참조링크와 같이 Item 노드 및 부노드가 될 수 있다. 이동링크는 사용자에게 다른 노드로 이동한다는 것을 알릴 수 있게 화살표를 가진 아이콘(icon)을 사용하였으며 이동 후에 다시 원래의 노드로 돌아 올 수 있는 기능도 시스템에서 제공하고 있다. 그림 6은 본 시스템이 지원하는 링크의 이용 예들을 보여주고 있다.

때로는 한 근원앵커(source anchor)가 여러개의 목적앵커(destination anchor)를 가지는 경우가 있는데 이를 지원할 수 있게 하기 위하여 본 시스템에서는 다중링크(multiple link)를 제공하고 있다. 성경에서와 같이 한 구절에 대한 관련절이 다수 있을 경우 이들 관련절로의 링크를 모두 텍스트상에 구현할 경우 사용자에게 혼란을 줄 수 있다. 따라서 본 시스템에서는 오직 하나의 링크만을 사용자에게 제시하되, 사용자가 이 링크를 선택하면 작은 윈도우를 이용해 여러개의 목적앵커를 제시하고 사용자의 요구에 따라 각 목적앵커를 차례로 방문하거나 특정 앵커로 이동할 수 있게 해 주었다.

이들 링크는 이외에도 이미지링크(image link), 사운드링크(sound link), 그래픽링크(graphic link)등이 있다. 이들은 그 기능상 참조링크나 이동링크에 속하지만 사용자에게 목적노드의 내용을 미리 알려주는 역할을 할 수 있다. 이미지링크는 한 이미지 안에 앵커영역(anchor region)을 가질 수 있어 마우스가 그위를 지나가면 나타나게 되어있다(그림 7 참조). 물론 사용자가 모든 이미지링크를 보고자 할 경우 토글 버튼(toggle button)을 이용할 수도 있다. 이미지의 앵커영역 역시 내부에 다른 앵커영역을 가질 수 있는데 이를 지원하기 위해서 앵커영역사이에 계층구조(hierarchical structure)를 구성한다. 즉 같은 자매(sibling) 앵커영역은 동시에 사용자에게 나타날 수 있으나, 상하관계에 있는 앵커영역은 동시에 나타날 수 없으며 사용자가 앵커영역의 레벨(level)을 변화시켜야만 하위의 앵커영역을 사용할 수 있다.

그림 7. 지도에 적용된 앵커영역의 구성 예

4-3. 탐색항해와 브라우징

사용자는 계층적 그룹(hierarchical group)을 이용하여 시작점을 찾을 수 있다. 계층의 개념은 일반적인 텍스트의 구성시에 흔히 쓰이는 개념으로 사용자가 처음 접하는 시스템에서 노드간의 계층적 관계를 빠른 시간내에 파악할 수 있다는 장점을 가진다.

이러한 계층을 형성할 수 있게 하기 위하여 본 시스템에서는 앞에서 서술한 바와 같이 Collection 노드를 사용하여 독립적인 일반 노드를 하나의 군으로 묶어준다. 형성된 계층정보는 계층적 뷰(Hierarchy view)를 통해 사용자에게 전달되며 사용자는 탐색의 어느 시점에서도 이 정보를 이용할 수 있다 (그림 11 참조). 또한 사용자는 탐색(searching)을 이용하여 시작점을 찾을 수도 있다. 본 시스템은 여러가지 탐색 방식을 지원하고 있다. 사용자는 검색된 노드내에서 탐색어를 가진 부분의 내용을 즉시 확인할 수 있으며 선택된 노드로 이동할 수 있다. 또 하나의 시작점 획득방식으로는 노드의 제목(title)을 이용하여 사용자가 알고 있는 노드로 즉시 이동하는 방식이다. 일반적으로 사용자는 한번 방문한 노드를 다시 방문하는 경우가 많은데 이때 사용자가 획득한 그 노드에 대한 정보를 시작노드의 획득에 이용할 수 있게 하였다.

사용자는 시작노드에서 다음노드로의 이동시에 여러가지 정보를 이용할 수 있다. 우선 가장 쉽게 이용할 수 있는 것이 지역 맵(Local map)이다(그림 11 참조). 본 시스템에서는 어떤 노드와 밀접하게 연관되어 있는 노드를 지역 맵에 나타나도록 하였는데, 예를 들어 그 노드에 관계된 도표나 사진 혹은 좀 더 자세한 설명이 나와 있는 다른 노드 또는 유사한 내용의 노드등이 이 지역 맵에 나타난다.

지역 맵에 나타나는 노드들은 저자가 직접 지정해 주어야하기 때문에 다량의 텍스트 정보를 하이퍼미디어화 할 때 사용자가 원하는 모든 관련 노드를 효과적으로 나타내기가 쉽지않다. 이럴 경우 오히려 사용자는 원하는 노드가 없는 것으로 간주하거나 방향상실에 빠지기 쉽다. 그러므로 관련된 노드를 저자가 아닌 시스템에서 자동적으로 연결시켜 줄 필요가 있다. 본 시스템은 각 노드의 내용을 분석하고 키워드(keyword)를 선정하여 노드간의 유사도(similarity level)를 계산한후 클러스터를 자동적으로 형성시켜주었다. 사용자에게 지역 맵이외에도 클러스터 뷰(Cluster view)를 통해 현 노드와 관련된 노드에 관한 정보를 획득할 수 있게 해 줌으로써 저자의 오류를 어느 정도 보완해 주도록 하였다.

본 시스템에서 각 지역 맵은 서로 다른 클러스터 부류(cluster class)로 이동시켜주는 역할을 할 수 있으며 클러스터 뷰는 관련된 노드의 집합을 보여주는 역할을 담당하기도 한다. 이렇게 함으로써 여러개의 관련 집합들을 효율적으로 묶어 줄 수 있으며 사용자는 관련도가 높은 노드는 클러스터 뷰를 통해서 얻을 수 있고 새로운 관련 클러스터 부류는 지역 맵을 이용하여 얻을 수 있다.

탐색시 사용자는 검색을 통해 새로운 노드에 접근할 수도 있는데 본 시스템에서는 전문색인(full-text index)을 이용하여 불리안 검색(Boolean retrieval) 및 확률 검색(probablistic retrieval)을 지원하고 있다. 그림 8은 탐색 윈도우를 보여주고 있다. 불리안 검색은 원하는 노드에 관해 정확히 기술할 수 있도록 해 주며, 확률 검색은 비전문가가 쉽게 검색을 행할 수 있게 해준다. 또한 본 시스템에서는 각 탐색어 간의 거리(proximity)를 이용하여 검색을 할 수 있도록 해준다. 거리는 물리적으로 주어질 수도 있으며, 같은 노드 혹은 같은 부노드에 나타나는 경우만을 검색대상으로 제한시킬 수도 있다. 이와 같이 탐색어 간의 거리를 제한 시킴으로써 부적합한 문헌을 제거시키고 정확도를 높일 수 있다. 본 시스템에서는 각 단어사이에 존재하는 바이트수를 두 단어 사이의 물리적 거리로 사용한다. 탐색된 결과는 사용자의 요구에 따라 탐색어를 빈도수 혹은 제목순으로 나열되며, 때로는 노드의 ID순으로 나열되기도 하는데 이는 성경과 같이 대부분의 사람들이 그 노드의 순

서를 잘 알고 있는 경우에 효과적으로 사용될 수 있을 것이다[1, 14].

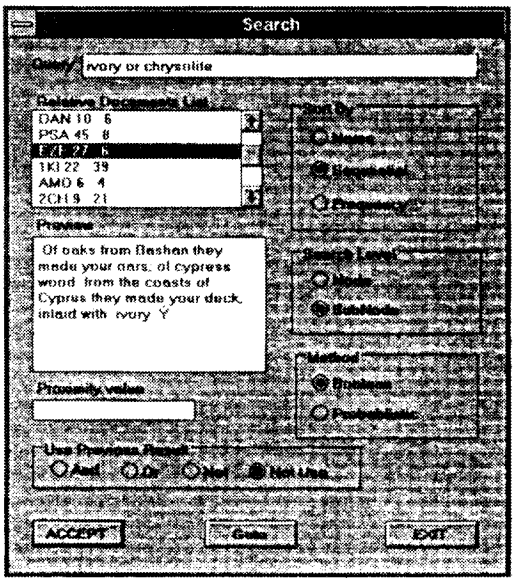

그림 8. 탐색 윈도우

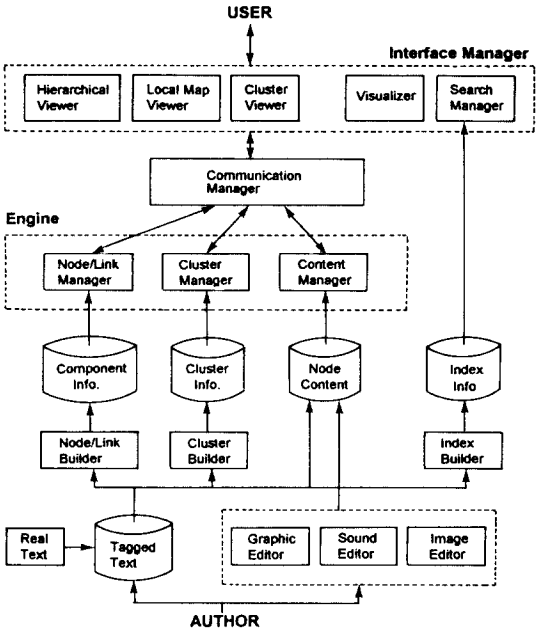

그림 9. 시스템 구성도

4-4 하이퍼텍스트 엔진 및 도구(tool)의 구성

그림 9는 본 시스템의 전체구성도를 보여주고 있다. 노드/링크 구성기(Node/Link builder)는 태그된 텍스트(tagged text)로부터 노드 및 링크정보를 추출하여 노드와 링크를 구성한다.

위의 과정을 통하여 생성되는 정보는 하이퍼미디어 엔진에서 사용하게 되는데 본 시스템의 엔진은 크게 노드 및 링크의 구조적관계를 관리하는 노드/링크 관리자(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 보여주는 기능을 담당하고 있는데 노드와 링크에 관련된 정보는 노드/링크 관리자를 통해 얻게되며 그에 따른 실제의 내용은 내용 관리자가 관리한다. 계층적 뷰는 Collection 노드의 정보를 이용하여 각 노드간의 계층적 구조를 사용자에게 보여주는 기능를 가진다. 지역맵 뷰는 각노드의 내용을 보여주는 표현기에 의해 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 경로관리자(History manager), 책갈피 관리자(Bookmark manager)등도 상위모듈에 속하며 이들이 사용하는 정보는 통신 관리자

에 의해 유지된다.

앞에서 사용되는 각종 정보를 구성하기 위하여 본 시스템에서는 여러가지 도구(tool)를 사용한다. 우선 영상에디터(Image editor)는 실제적인 영상의 재구성 기능을 가지고 있지 않으며 다만 영상내에서 사용되는 앵커영역(anchor region)을 구성하고 각 앵커의 계층적 관계를 형성시켜준다. 각 앵커영역은 태그를 통해 실제의 목적노드와 연결된다. 따라서 영상에디터에 의해 구성되는 앵커영역은 항상 근원앵커(source anchor)로만 작용한다. 그래픽에디터(Graphic editor)는 각각의 그래픽 객체를 형성하고 수정 및 저장할 수 있는 기능을 갖는다(그림 10 참조). 이때 각 객체는 개별적으로 이동, 변환, 수정, 삭제될 수 있어서 손쉽게 그래픽을 구성할 수 있다.이렇게 형성된 자료는 태그된 텍스트를 통해 서로 연결된다. 색인 구성기는 태그된 텍스트에 나타나는 각각의 단어들을 추출하고 불용어(stop word)를 제거한 후 도치화일(inverted file)로 저장한다. 현재 색인 구성기로 약 5M바이트 정도의 텍스트 화일의 전문색인을 구성하는데 약 3M 바이트 정도를 소요하며 시간은 PC 486급으로 30분 가량 소요된다.

4-5. 사용자 인터페이스(user interface)

그림 11에서 왼편의 윈도우는 탐색항해를 위해 사용되는데, 위의 그림은 계층 뷰를 보여주고 있으며 아래의 그림은 지역 맵을 보여준다. 아래 그림의 지역 맵은 사용자의 선택에 따라 클러스터 뷰로 대체될 수도 있으며 다른 노드로 이동할 경우 자동적으로 그 노드에 관계된 내용으로 변경된다.

주 메뉴의 기능으로는 화면의 윈도우들을 재배치할 수 있는 뷰 기능과 경로 추적기(Path history) 및 책갈피(Bookmark) 윈도우를 띄울 수 있는 기능 및 탐색 윈도우 생성기능, 노드 파인더(Node finder)를 보여주는 기능등이 있다. 이들중 많이 사용되는 기능은 그림 11의 위쪽에 있는 리본 윈도우의 버튼을 통해 연결시켜서 쉽게 찾아쓸 수 있게 되어있다.

브라우징 윈도우나 리본 윈도우는 사용자의 선택에 따라 화면에 나타나기도 하며 사라지기도 하므로 사용자의 용도에 따라 화면을 마음대로 재배치시킬 수 있다. 또한 각 윈도우는 크기를 마음대로 변화시킬 수도 있으며, 스크롤(scroll) 기능을 제공하기 때문에 노드의 크기에 구애받지 않고 사용할 수 있다.

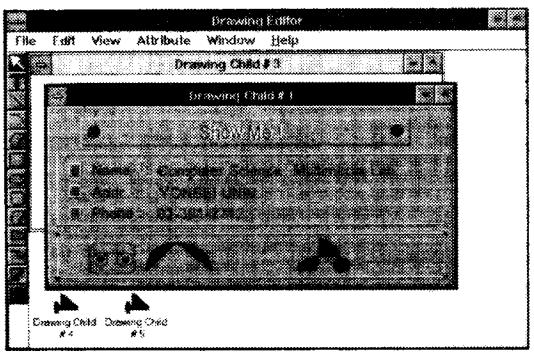

그림 10. 본 시스템의 그래픽에디터를 이용한 그래픽 작성의 예

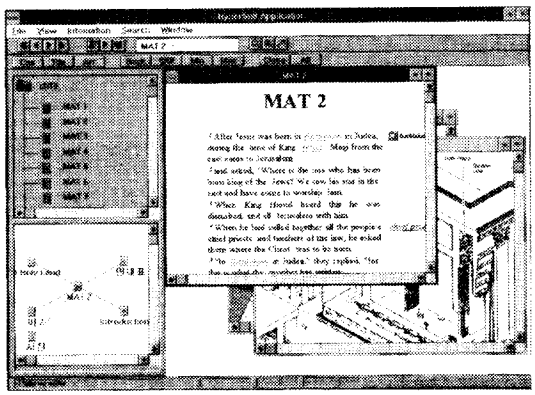

그림 11. 본 시스템을 이용한 응용의 예

V. 결 론

사용자가 원하는 정보에 쉽고 효율적으로 접근할 수 있도록 하기 위하여 본 시스템에서는 계층적 뷰, 지역 맵, 클러스터 뷰 등을 지원하고 있다. 이외에도 다양한 링크와 여러가지 종류의 노드 및 탐색항해 지원도구와 검색 시스템을 제공하고 있다. 이러한 탐색항해 지원시스템들이 효과적으로 활용될 수 있는지를 판단하기 위해서는 적절한 적용 분야를 찾아 응용 시스템을 구축하여야 할 필요가 있다. 이에 본 연구실에서는 성경 및 백과사전에 본 시스템을 적용하기로 결정하고 이미 성경의 경우 전문을 입력하였으며 현재 링크을 구성하는 작업을 진행하고 있다.

현재 본 시스템은 독자적인 하부저장 시스템을 갖지 못하고 있는데 이는 큰 응용시스템을 구축하는데 많은 애로점을 발생시키고 있다. 또한 텍스트의 포맷 지정을 위한 워드 프로세서 기능과 각 도구(tool)간의 인터페이스와 통합적인 작업환경을 위하여 통합 에디터의 개발이 요구된다. 또한 응용시스템의 구축과정에서 발생한 여러가지 문제점을 보완할 수 있도록 본 시스템을 지속적으로 수정, 확대해 나갈 예정이다.

참 고 문 헌

[1] 신미영, "효율적인 정보 검색을 위한 클러스터 기반의 하이퍼텍스트 설계", 석사학위 논문, 연세대학교, 1992.

[2] C.J. van Rijsbergen, Information Retrievel, BUTERWORTHS, 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. 57－67, 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 Ne In : Set Based Hypermedia for Taxonomic Reasoning", Hyertext '91 Proceeding, pp. 233－242, Dec. 199.

[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] Pater A. Gloor, "CYBERMAP–Yet Another Way of Navigation in Hyperspace", Hypertext '91 Proceeding, pp. 107–122, Dec. 1991.

[11] Walter M., "IRIS Intermedia : Rushing 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 Process-

ing, Addison-Wesley, 1989.
[15] Shneiderman b., Designing the User Interface, Second edition, Addison-Wesley, 1992.