

# 탐색과 브라우징을 지원하는 하이퍼미디어 시스템의 설계

## Design of a Hypermedia System for Effective Searching and Browsing

고영곤(Young-Kon Ko)\*

최윤철(Yoon-Chul Choy)\*

□ 목

차 □

- |                                   |                        |
|-----------------------------------|------------------------|
| 1. 서론                             | 4.2 하이퍼미디어 시스템의 구성     |
| 2. 하이퍼텍스트 시스템의 발전                 | 4.3 시스템의 구현환경          |
| 3. 탐색과 브라우징                       | 5. 적용의 예-성경            |
| 4. 하이퍼미디어 시스템의 설계                 | 5.1 성경 시스템 구축의 개요      |
| 4.1 탐색 및 브라우징을 위한 정보공간의<br>구성과 기능 | 5.2 성경 시스템에서의 탐색과 브라우징 |
|                                   | 6. 결론                  |

### 초 록

하이퍼미디어 시스템은 멀티미디어 정보간의 연상적 관계를 링크, 노드개념에 입각하여 지원함으로써 기존의 데이터베이스 시스템이나 텍스트 검색 시스템의 한계성을 극복해 준다. 본 연구는 텍스트, 그래픽, 영상, 목소리/사운드 등의 멀티미디어 정보를 지원해 주는 하이퍼미디어 시스템을 설계, 구현하였고 시스템의 기능을 보여주기 위하여 성경과 성경 관련 정보에 적용하여 응용시스템을 구축하였다. 특히 본 시스템에서는 탐색과 브라우징 기능을 효율적으로 통합함으로써 하이퍼미디어 시스템의 정보검색 기능과 사용자 인터페이스를 높이도록 설계하였다.

### ABSTRACT

Hypermedia system supports associative linking concept for multimedia information using link and node concept, and overcomes the limitations of database system and text retrieval system in some application areas. This study shows the design and implementation of a hypermedia system which supports text, graphics, image and voice/sound information. This system has been designed to integrate the browsing and searching functions of the hypermedia system for efficient multimedia information retrieval and user-interface. To demonstrate the function and capability of the system, an application was made in the area of Bible and related information.

\*연세대학교 전산학과

## 1. 서 론

1960년대 이후 컴퓨터 기술의 급속한 발전에 힘입어 데이터베이스 시스템 응용이 매우 확산되어 왔다. 특히 지난 십수년동안 관계 데이터베이스 기술이 성숙하면서 데이터베이스 응용은 매우 안정된 기술로서 관계 데이터베이스 시스템이 폭 넓게 활용되고 있다. 다른 한편으로는 1960년대 들어서 컴퓨터가 정보검색 분야에 본격적으로 이용되기 시작하면서 최근까지 많은 발전을 거듭해 왔다. 이것은 컴퓨터 하드웨어의 급속한 발전과 소프트웨어 기술의 향상 및 응용에서 쌓인 경험들이 큰 역할을 했을 것으로 생각된다. 그러나 최근 이러한 발전과 더불어 지금까지 구조화되고 정형화(structured and formatted)된 데이터 형식을 지원하는 데이터베이스와 텍스트 검색을 주로 다루는 정보검색의 한계성을 벗어나 멀티미디어 정보를 효율적으로 저장, 처리, 검색할 수 있는 정보 시스템의 필요성이 광범위하게 요구되고 있다. 또한 1945년 Vannevar Bush에 의하여 제창된 하이퍼텍스트(hypertext) 개념이 1980년대 들어와서 멀티미디어 기술의 발전과 더불어 빠른 속도로 그 유용성과 기술적 가능성이 증명되었으며 미국을 중심으로 많은 시스템이 개발되어 왔다. 하이퍼텍스트 시스템은 서로 관련있는 정보들을 링크로서 연결하여 브라우징 함으로써 매우 획기적인 정보검색 기법으로 인식되고 있다. 종래의 정보시스템과는 달리 하이퍼텍스트는 연관적인(associative) 인간의 사고(human cognitive) 과정과 유사하게 정보를 비순차적(nonsequential)적으로 상호 연상되는 정보를 링크를 따라 접근하게 된다. Nielsen[10]은 하이퍼텍스트 정보 시스템이 1990년대 말 경에는 매우 광범위하게 개발되어

이용될 것으로 예측하고 있다. Ted Nelson은 21세기 초에는 세계의 수 많은 정보가 상호 연관되어 몇개의 매우 큰 총체적인 하이퍼텍스트(global hypertext) 형태로 개발될 것을 이야기하고 있다.

이러한 측면에서 볼때 정보화 사회에서 하이퍼텍스트 또는 하이퍼미디어 시스템의 개발은 매우 중요하고 획기적인 발전을 초래할 수 있는 기술로 인식되고 있다.

## 2. 하이퍼텍스트 시스템의 발전

1945년 Vannevar Bush가 제안한 Memex 이후 1960년대에 들어서면서 하이퍼텍스트 시스템이 설계, 구현되기 시작하였다. 초기 시스템으로는 Engelbart의 Augment/NLS, Nelson의 Xanadu, Brown 대학의 Fress 시스템들을 들 수 있다 [2].

1980년대에는 대학, 연구소를 중심으로 많은 하이퍼텍스트 시스템이 개발되었는데 그 응용분야 목적에 따라 기능이 매우 다양하다.

Conklin[2]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, 문제 해결 분야, 프로그래밍 개

발 분야에서 이용된다.

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

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

### 4) 실험적 하이퍼텍스트 시스템

몇가지 응용분야를 염두해 두고 하이퍼텍스트 기술의 실험적 응용을 위해서 개발된 시스템으로 XEROX RARC의 NoteCard, IRIS의 Intermedia, Tektronix의 Neptune 등이 이 부류에 속한다.

하이퍼텍스트가 모든 정보검색 분야에 적합한 것은 아니며 Ben Shneiderman[14]은 하이퍼텍스트 시스템 분야에 적합한 응용분야를 결정하기 위하여 다음의 원칙을 판단 기준으로 제시하였다.

- 대량의 정보가 다수의 정보 단위로 구성되어 있어야 한다.
- 정보 단위들은 상호 연관관계가 있어야 한다.
- 사용자는 한 시점에서 대체적으로 매우 작은 분량의 정보만을 필요로 한다.

이러한 원칙에 입각하여 볼 때 하이퍼텍스트에 가장 적합한 응용분야로는 전자사전, 전자 백과사전, 교육 분야, 참고문헌 분야, 의학정보 분야, 온라인 분야, 소프트웨어 공학(CASE), 전자출판, 외국어 교육등을 들 수 있다.

하이퍼텍스트의 많은 장점에도 불구하고 하이퍼텍스트의 자유로운 탐색에 기인하는 하이퍼텍스트

공간(hyperspace)에서 사용자 방향상실(user disorientation)이 문제점으로 부각되고 있다. 이 문제를 해결하기 위한 기법으로 전체구성도(overview diagram), 경로 추적기(path history), 안내지침(guided tour), 책갈피(bookmark), 탐색 도구(searching tool), 계층 구조(hierachical structure) 등이 하이퍼텍스트에 따라 선별적으로 지원되고 있다[7, 11, 17].

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 탐색(searching) 기능은 부족한 편이다. 따라서 교육용 소프트웨어 제작, 전시용 멀티미디어 시스템(presentation multimedia system), 안내 시스템 등의 제한된 범위 내에서 매우 효과적으로 이용될 수 있다.

하이퍼텍스트 시스템의 응용분야가 매우 다양하여 하나의 일반적인 하이퍼텍스트 엔진(hypertext engine)으로는 모든 분야에 응용하기에 부적합한 측면이 있다. 본 연구에서 개발한 하이퍼미디어 시스템은 앞의 분류에 의하면 구조적 브라우징 시스템과 실험적 하이퍼텍스트 시스템 부류에 속하는 시스템으로 특히, 백과사전, 참고문헌 정보, 전자사전(electronic dictionary), 전자책(electronic book) 등에 활용할 목적으로 개발하였다. 특히 하이퍼텍스트 시스템의 가장 큰 문제점인 방향 상실을 해결하기 위하여 브라우징(browsing)과 탐색(searching) 기능이 통합되도록 시스템을 설계하였다. 항해탐색(navigation) 장치로 계층적 뷰(hierarchical view), 지역 맵

(local map), 경로 추적기(path history), 책갈피(bookmark) 기능을 지원하였다.

### 3. 탐색과 브라우징

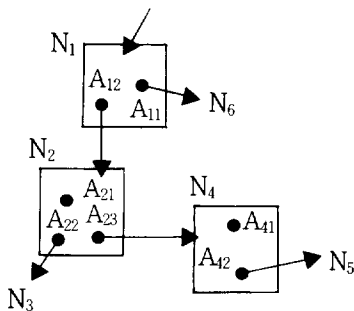
일상적으로 접하게 되는 정보란 정보 조각들의 연속적인 묶음으로 나타난다. 이렇듯 정보를 이루는 한 단위를 정보단위(information unit)라고 볼 수 있으며, 각 정보단위를 의미적으로 연관시켜 놓은 영역을 정보공간이라 한다[11]. 이때 각 정보단위 뿐만 아니라 정보단위 간의 연결 관계 역시 지식(knowledge)을 내포하고 있다고 볼 수 있는데, 결국 정보검색이란 정보공간내의 지식을 이용하여 원하는 정보단위로 접근하는 행위라고 말할 수 있다.

고전적인 하이퍼텍스트 시스템의 정보공간은 정보단위를 나타내는 노드와 각 노드 간을 연결하는 링크로 구성된다. Parunak [8]은 하이퍼텍스트 시스템을 정보공간의 구성 방식에 따라 분류할 것을 제안하고 있는데, 그의 분류 방식을 따른다면 고전적인 하이퍼텍스트 시스템은 그래프식(graph-based) 시스템으로 분류될 수 있을 것이다. 그래프식 시스템에서 정보단위 간의 의미적 연결은 링크로의 연결을 통해 이루어진다. 사용자는 링크를 통해 표시되는 노드간의 연관 관계에 대한 지식을 이용하여

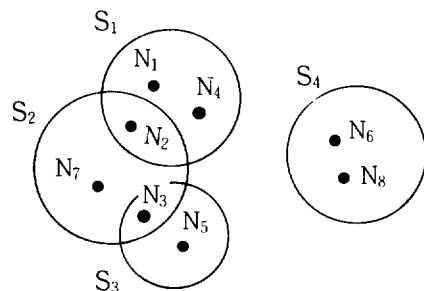
목적노드로 접근해 갈 수 있다. 즉 [그림 1]에서와 같이 노드 N1에서 노드 N5로 이동할 경우 사용하는 노드 N1의 앵커 A12를 이용하여 노드 N2로 이동한 후 다시 앵커 A23를 이용하여 노드 N4에 간 후 다시 노드 N5에 도달할 수 있게 된다.

반면에 기존의 검색 시스템은 집합식(set-based) 시스템에 가까운데, 이러한 집합식 시스템에서 정보공간은 연관된 정보단위들을 클러스터(cluster)로 묶음으로써 구성된다. 이때 각 정보 단위는 여러 클러스터에 중복적으로 포함되어지게 된다. 이러한 시스템에서의 탐색항해(navigation)는 [그림 2]에서 보듯이 클러스터 내의 노드를 선택하고 그 노드를 포함하는 클러스터 중 다른 하나를 선택하는 행위를 반복함으로써 이루어진다.

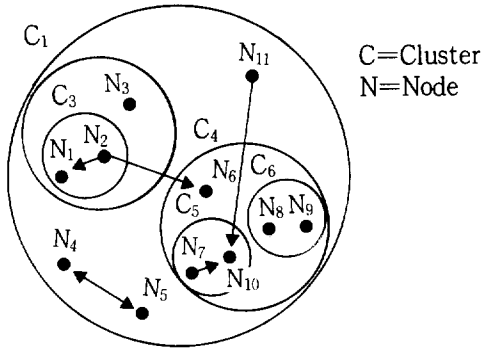
결국 그래프식 시스템에서는 연관된 정보에의 접근이 노드간의 연결관계에 지나치게 의존되고 있어서 오히려 연관 정보로의 접근을 방해할 소지가 있는 반면에 집합식 시스템에서는 서로 다른 부류로의 이동이 쉽지 않은 단점을 가진다. 그러나 각 정보공간 구성방식은 나름대로의 장점을 가지고 있기 때문에 이들 정보공간을 보완적으로 통합할 경우 연관성이 높은 노드로의 이동뿐만 아니라 다른 부류의 클러스터 집합으로의 이동도 용이한 정보공간이 형성될 수 있을 것이다.



[그림 1] 그래프 기반 모델에서의 정보공간 구성도



[그림 2] 집합 기반 모델에서의 정보공간 구성도



[그림 3] 통합 모델에서의 정보공간 구성도

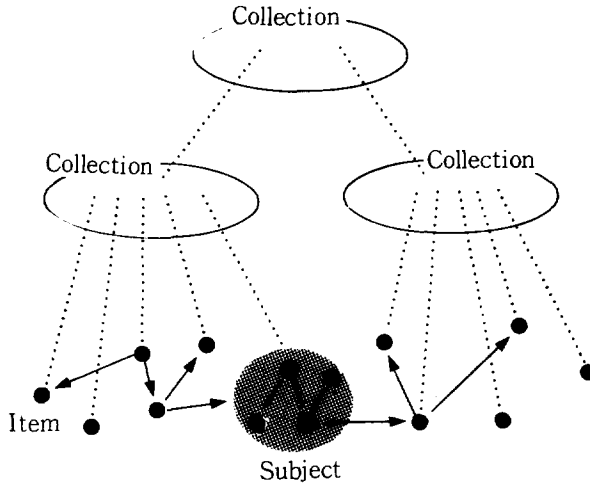
[그림 3]은 두 개의 정보공간을 단순 통합하여 구성한 정보공간을 보여주고 있다. 이때 통합된 정보공간에서의 노드 N1에서 N3로의 이동은 N1과 N3가 같은 부류에 속하기 때문에 쉽게 이루어질 수 있다. 또한 클러스터 C2에서 C4로의 이동도 N2에서 N6로의 링크를 이용해 쉽게 이동할 수 있다. 즉 두 정보공간의 통합은 가까운 거리의 정보들은 집합을 통해 파악할 수 있으며 링크는 다른 부류의 정보군으로 이동할 때 쓰일 수 있다는 장점을 가진다. 본 연구에서는 이러한 개념에 입각하여 하이퍼텍스트의 브라우징 방식과 클러스터링 방식을 통합함으로써 보다 효율적인 탐색항해가 가능하도록 설계하였다.

#### 4. 하이퍼미디어 시스템의 설계

탐색항해(navigation)를 위한 기본적인 접근방법에는 내용을 통한 탐색과 정보공간의 구조를 통한

탐색이 있다. 지금까지 진행되어 온 하이퍼미디어 시스템에서의 탐색항해에 관한 연구는 주로 구조적 탐색에 집중되어 왔는데, Halasz[9]에 의하면 이러한 구조적 탐색은 저자(author)에 의해 인위적으로 구성된 노드 및 링크정보를 통해 가장 적절하게 이루어진다. 그러나 방대한 양의 텍스트를 기반으로 하는 정보를 하이퍼미디어화한 경우에도 이러한 방식이 충분한 효과를 발휘할 수 있으리라고는 기대하기 힘든데, 이는 시스템에 접근하는 이용자의 사용의도가 다양하기 때문에 이를 모두 수용할 수 있는 적절한 정보공간을 구현하기가 어렵기 때문이다.

이에따라 본 시스템은 방대한 텍스트 기반 정보의 하이퍼미디어 시스템화를 위해 이용자의 탐색을 위한 각 노드와 링크의 구조적 연결관계를 이용한 탐색뿐만 아니라 정보의 내용을 이용한 탐색항해를 위해 각 노드간의 내용상의 유사성을 고려한 클러스터를 형성하여 탐색에 이용할 수 있도록 하였으며, 전문색인(full-text index)을 통해 색인어의 거



[그림 4] 계층 그룹(hierarchical group)의 구성 방식

리(proximity)를 고려한 다양한 검색을 지원하고 있다.

#### 4.1 탐색 및 브라우징을 위한 정보공간의 구성과 기능

##### 4.1.1 시작노드(source node)의 선정

원하는 정보를 획득하기 위해서는 효율적인 시작점을 제시해줄 수 있는 방법이 필요하다. 본 시스템에서의 시작점 획득은 세가지 방식으로 수행할 수 있다.

첫째 방식은 계층적 그룹(hierarchical group)을 이용하여 시작점을 찾는 것이다. 계층의 개념은 일반적인 텍스트의 구성시에 흔히 쓰이는 개념으로 사용자에게 처음 접하는 시스템에서의 노드간의 관계를 빠른 시간내에 알려줄 수 있는 장점을 가진다. 이러한 계층을 형성할 수 있게 하기 위하여 본 시스템에서는 Collection 노드를 사용한다. Collec-

tion 노드는 일반 노드와는 달리 내용을 내포하지 않는 노드로서 독립적인 일반 노드들을 하나의 군으로 묶어주기 위한 정보만을 가진다. 형성된 계층 정보는 계층적 뷰(Hierarchy view)를 통하여 사용자에게 전달되며 사용자는 탐색의 어떤 시점에서도 이 정보를 이용할 수 있다. [그림 4]는 본 시스템에서의 Collection 노드와 기타 노드와 계층관계를 보여 주고 있다.

둘째로 사용자는 검색(searching)을 통해 시작점을 획득할 수 있는데 이를 위해서는 사전에 탐색하고자 하는 노드에 대한 예비지식이 요구된다. 사용자가 입력한 탐색어는 불리안 검색 또는 확률 검색을 통해 탐색어를 가진 모든 노드 또는 부노드(subnode)를 찾아온다. 사용자는 검색된 노드내에서 탐색어를 가진 부분의 내용을 즉시 확인할 수 있으며 선택된 노드로 이동할 수 있다.

또 하나의 시작점 획득방식으로는 노드의 제목(title)을 이용하는 방식이 있다. 일반적으로 사용

자는 한번 방문한 노드를 다시 방문하는 경우가 많은데 이때 사용자가 획득한 그 노드에 대한 정보를 시작노드의 획득에 사용할 수 있게 하는 것이 바람직할 것이다.

#### 4.1.2 이동을 위한 정보구조 및 기능

사용자는 탐색시에 여러가지 정보에 접하게 되며 이들 정보에 근거하여 다음 탐색 노드를 결정하게 된다. 하이퍼미디어 시스템에서 이러한 정보로 가장 자주 이용하게 되는 것이 링크이다. 본 시스템에서는 각 노드간의 구조적 연결 관계를 알려주는 링크와 연결된 노드의 내용 정보를 인지시킬 수 있는 링크 및 수행방식을 보여주는 링크를 가지고 있다. 전자의 링크로는 계층적 그룹을 형성시키기 위한 계층적 링크(Hierarchy link), 현 노드의 부수적 정보 또는 관련 정보를 보여주는 지역 링크(Local link) 등이 있다. 이때 계층적 링크는 Collection 노드에 내포되어 있으며, 지역 링크는 현 노드 내용의 일부로 구성되어 있다. 이들 링크는 각 노드의 연결 관계를 보여줄 뿐이며 연결된 노드의 내용에 관련된 정보는 거의 가지지 않는다. 본 시스템은 내용과 관련된 링크로서 영상 링크(Image link), 사운드 링크(Sound link), 이동 링크(Move-to link), 참조 링크(Reference link), 핫링크(Hot link), 다중 링크(Multiple link) 등을 지원하는데 이들은 연결된 노드의 내용을 추측할 수 있게 해 줄 뿐만아니라 그 수행방식에 대해서도 알 수 있게 해준다.

핫 링크는 특정 단어에 혹은 구절에 대한 간략한 설명을 보여준다. 이 핫 링크의 기능을 지원해주기 위하여 본 시스템은 정보사전을 구성하여 사용하고 있다. 참조 링크는 같은 노드간의 연결을 지원하는

링크로써 현 노드를 그대로 유지시켜준 상태에서 다른 노드의 내용을 볼 수 있게 해 준다. 이동 링크는 현 노드(current node)의 내용을 목적 노드(destination node)의 내용으로 대체시켜 준다. 다중 링크는 여러개의 목적앵커(destination anchor)를 가진 링크로써 이동 링크 및 참조 링크의 기능을 모두 가질 수 있다.

사용자는 또한 클러스터 정보를 이용하여 이동을 할 수 있다. 현재 선택된 노드와 내용상 관련된 노드들을 나열시켜주는 클러스터 뷰(Cluster view)를 이용하여 사용자는 자신이 원하는 노드를 선택할 수 있다. 이때 클러스터 정보는 각 노드간의 구조적 관계가 아니라 내용적 연관 관계를 보여주기 때문에 링크와는 전혀 다른 방식으로 구현되지만 실제로는 링크와 유사한 기능을 한다.

이동을 도와주는 것으로 이들 외에 경로 추적기(Path history)와 책갈피(Bookmark)가 있다. 이들은 사용자의 탐색경로를 보여주고 표시해둔 정보를 유지시켜줄 뿐만 아니라 선택을 통해 다시 노드에 접근하는 것을 가능하게 해준다.

#### 4.1.3 검색을 위한 정보구조 및 기능

본 시스템은 검색을 위해 전문을 색인어로 구축하는 방식을 사용한다. 이때 색인어로 선택되는 단어는 불용어(stop word) 목록에 포함되지 않아야 한다. 또한 관련어는 한 어근으로 바꾸어주는 작업을 한후 그 단어가 포함되어 있는 노드 및 노드내에서의 위치 정보와 함께 저장된다.

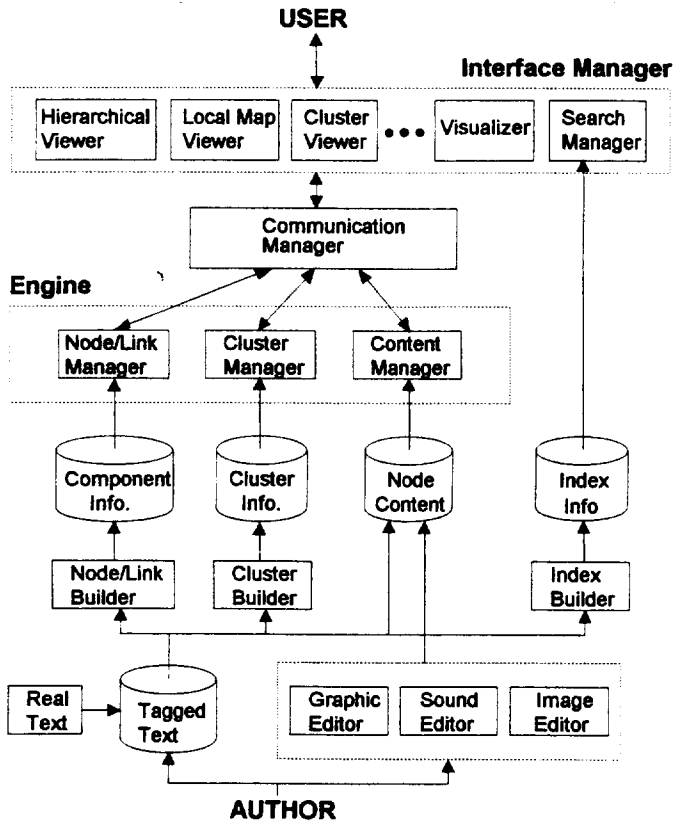
블리안 검색 및 확률 검색이 지원되며 각 단어의 위치에 따른 근거리(proximity) 검색이 가능하다. 또한 사용자는 검출된 결과를 노드단위 또는 부노드(subnode) 단위 등으로 선택할 수 있다.

### 4.2 하이퍼미디어 시스템의 구성

[그림 5]는 본 시스템의 전체구성도를 보여주고 있다. [그림 6]에서 보는 것과 같이 태깅(tagging)을 한 응용은 노드 구성기(Node /Link builder)를 통해 텍스트 내에서의 노드 및 링크정보가 추출된다. 영상 에디터(Image editor)는 실제적인 영상의 재구성 기능을 가지고 있지 않으며 다만 영상내에서 사용되는 앵커 영역(anchor region)을 구성하고 각 앵커의 계층적(hierarchical) 관계를 형성시켜준다. 그래픽 에디터(Graphic edi-

tor)는 각각의 그래픽 객체(graphic object)를 형성하고 수정 및 저장할 수 있는 기능을 갖고 있다. 이렇게 형성된 자료들은 위의 태그된 텍스트에 나타나는 각각의 단어들을 추출한 후 불용어(stop word)를 제거하여 도치화일(inverted file)로 저장한다. 현재 각 자료는 기존의 운영체제가 제공하는 파일 시스템을 그대로 이용하여 저장 관리하고 있는데, 앞으로 다양한 멀티미디어 자료를 효과적으로 관리하기 위하여 별도의 저장 관리 시스템을 개발하여 사용할 예정이다.

앞의 과정을 통하여 생성되는 데이터는 하이퍼



[그림 5] 전체 시스템의 구성도



```

{ \nodec 0015
  \type linear
  \title Hierarchics
  \text
  \loc 80 80 0 0
  \title 1
  { The structure of a hypertext database is a major factor that determines how
    easy it it to create, use and update.. }
  \text
  \loc 80 80 0 0
  \title 2
  { One way of organizing a hypertext database that simplifies <\ link hot in 0 >~browsing~1|102~ and
    authoring is a hierarchical structure.. }
  \text
  \loc 80 80 0 0
  \title 3
  { test version 5 of <\ link multi in 0 >~hypertext~0016|6~0017~0018~ applicaiton.
    the test version is application and hierarchical structure.. }
  \text
  \loc 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
  \loc 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~0016|3~
    <\ link sound out after 4 >~style~0020~
    <\ link hot out after 5 >~browsing~1|101~
    <\ link pict out after 5 >~hypertext~0016~0017~0018~
    <\ link goto out before 3 >~OxfordDictionary~0019~
    <\ link sound out before 4 >~style~0020~
    <\ 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~0016|8~
  }
  \keyword
  { hierarchy
    browse
    database
    hypertext
    structure
    organize
    authoring
  }
}

```

[그림 6] 태그된 텍스트

미디어 엔진(hypermedia engine)에서 사용하게 되는데 본 시스템의 엔진은 크게 노드 및 링크의 구조적 관계를 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 인터페이스를 하게된다.

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. 표현기는 내용 관리자로부터 정보를 얻는 유일한 모듈이다. 계층적 뷰는 Collection 노드의 정보를 이용하여 각 노드간의 계층적 구조(hierarchy)를 사용자에게 보여주는 기능을 가진다. 지역 맵 뷰(Local map viewer)는 각 노드의 내용을 보여 주는 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(Communication manager)에 의해 유지된다.

#### 4.3 시스템의 구현 환경

본 시스템은 PC Windows 환경에서 개발되었고 GUI(Graphical User Interface) 방식을 사용자

인터페이스로 제공하고 있다. 엔진 및 사용자 인터페이스 모듈은 객체지향적 언어(object-oriented language) C++을 이용하여 구현하였고, 기타 에디터 및 자료 구성기들은 C를 이용하였다.

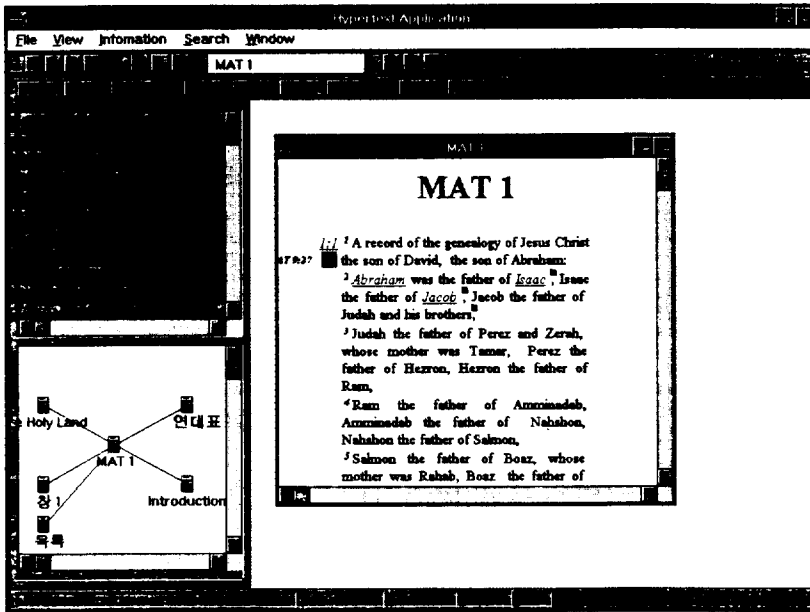
본 시스템을 이용하여 응용 시스템을 구축할 경우 각 응용 시스템당 본래의 태그된 텍스트보다 1.5배 가량의 자료 화일이 생성되며 색인의 경우 총 텍스트 크기의 절반 가량의 크기로 형성된다. 결국 한 응용시스템을 구축하는데 필요한 자료의 크기는 원래 크기의 두배 가량이 소요된다. 현재 이들 자료를 관리하기 위한 별도의 데이터베이스 시스템이나 화일 시스템은 구현하지 않았으며 기존의 운영체계를 이용하여 관리하고 있는데 앞으로 이들을 관리하기 위한 하부 저장 시스템을 개발하여 이용할 계획이다.

#### 5. 적용의 예 - 성경

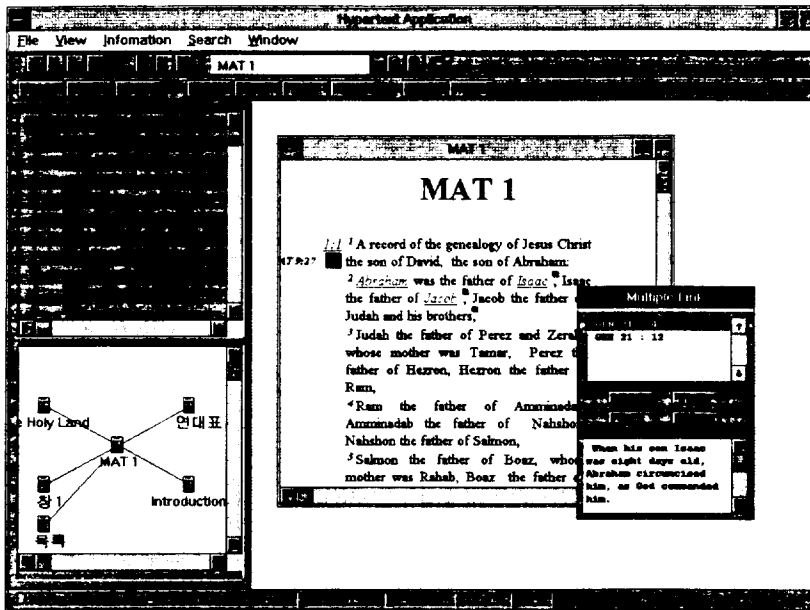
성경의 경우 방대한 양의 텍스트를 기본 정보로 가지고 있으며 많은 상호 참조(cross reference)를 가지고 있을 뿐만 아니라 그 종류도 다양하다. 성경은 신·구약 총 66개의 책(Book)으로 구성되고 다시 각 책은 최소 하나에서 최대 150개의 장(Chapter)으로 구분되어 있으며, 각 장은 30~40개의 절(Verse)을 가지고 있기 때문에 본 시스템의 특성을 보여 주기에 적절한 적용 분야로 판단된다. 이에 본 연구실에서는 영문 NIV(New International Version) 성경과 한글 성경의 전문을 본 시스템을 이용하여 하이퍼텍스트화 하였다.

##### 5.1 성경 시스템 구축의 개요

본 성경 시스템에서는 각 장(Chapter)이 하나의



[그림 7] 초기 화면



[그림 8] 다중 링크의 구현 모습

노드로 구성되었으며 각 절은 노드 내의 부노드(subnode)로 구성된다. 또한 각 책을 Collection 노드로 형성하여 계층 노드(hierarchical node)로 사용한다. 따라서 총 2887개의 노드가 형성되었는데 이들 노드 중에는 성경의 본문 이외에도 사진, 그래픽, 기타 텍스트 노드등이 포함되어 있다. 현재 이중 일부 노드 만이 시범(demonstration)을 위한 링크정보를 가지고 있으며 계속적으로 링크를 추가시켜 나가고 있다.

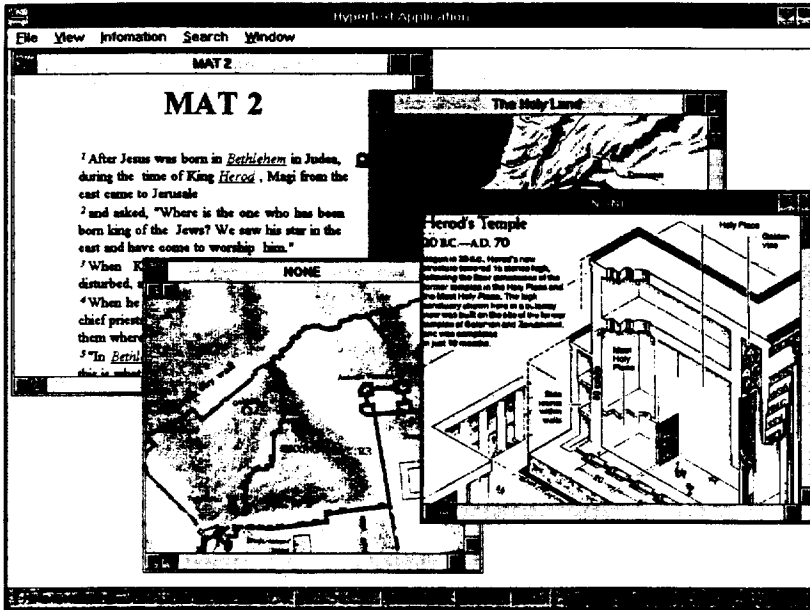
본 시스템은 영문 노드와 한글 노드가 함께 내재되어 있는데 이중 영문 노드만을 대상으로 전문 색인(full-text indexing)을 하여 약 5만개 가량의 인덱스를 구축하였다. 각 인덱스는 자신이 포함되어 있는 장, 절, 장내에서의 실제 위치(physical location)를 가지고 있다. 또한 의미있는 클러스터 정보를 추출해 내기위해 형성된 인덱스와 별도로 각 노드별로 10~20정도의 키워드를 추출해 낸후 이를 이용하였다. 현재 대상 노드가 방대하기 때문에 신약의 복음서를 우선 대상으로 선정하여 키워드 추출작업을 수행 하였다.

## 5.2 성경 시스템에서의 탐색과 브라우징

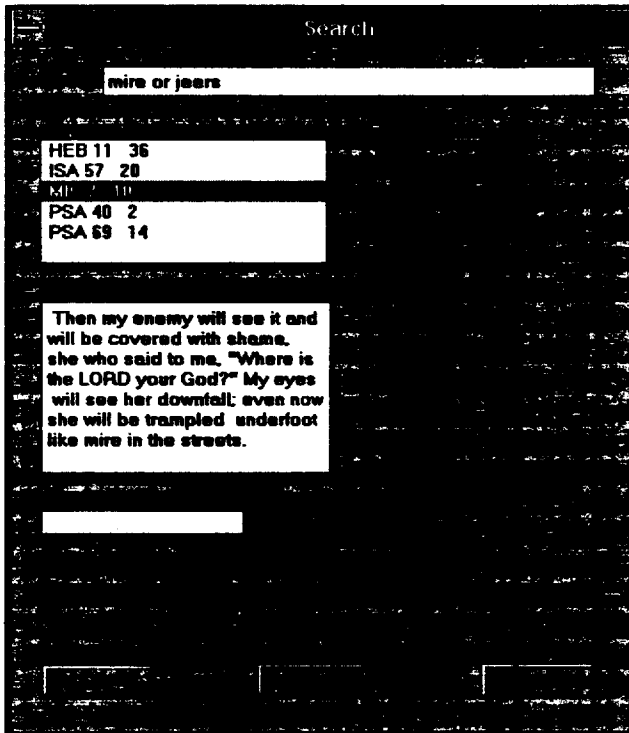
사용자가 성경 시스템을 작동시키게 되면 계층적 뷰를 처음으로 대하게 된다. 사용자는 성경의 구조에 대해 잘 알지 못할 경우라도 계층적 뷰를 통하여 전체 시스템의 노드 구조를 파악할 수 있게 되며 자신이 원하는 노드를 선정하여 탐색항해를 시작할 수 있다[그림 7]. [그림 7]은 본 시스템의 초기 화면이다. 만약 자신이 보고자하는 노드의 제목을 잘 알고 있을 경우 노드 파인더(Node finder)를 이용하여 직접 노드에 접근할 수도 있다. 또한 검색 시스템을 이용하여 노드에 접근할 수도

있는데 이 경우 탐색어를 내재하고 있는 장 또는 절들이 검색된다. 성경과 같은 응용분야에서는 각 주제별로 별도의 색인을 구축해야 할 필요가 있는데 아직 본 시스템에서는 지원해 주지 못하고 있다. [그림 10]은 본 시스템에서의 탐색 윈도우를 보여주고 있다. 탐색 윈도우는 사용자의 선택에 따라 탐색 방식, 거리(proximity) 정도, 결과의 출력 방식 등을 정할 수 있게 해준다.

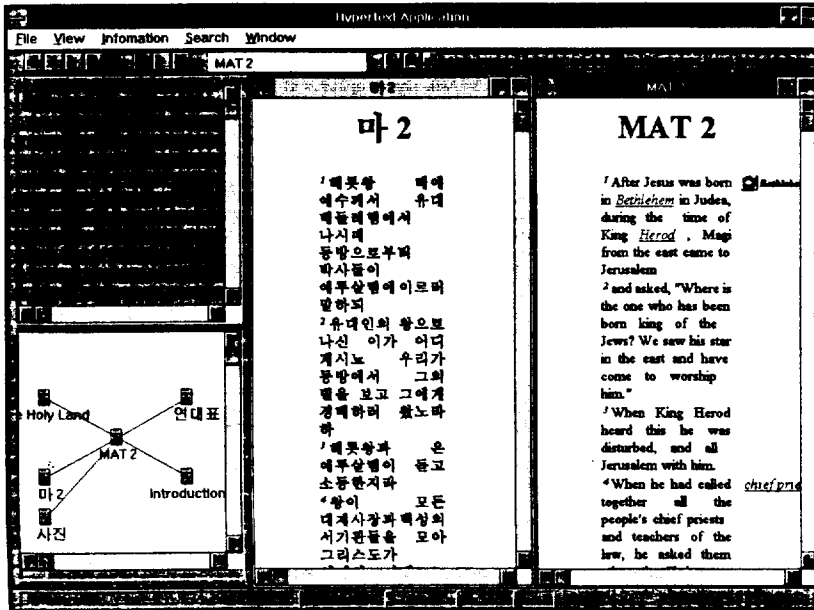
성경에서는 다양한 종류의 상호참조를 사용하고 있는데 가장 흔히 나타나는 것으로는 각절에 대한 주석과 특정 절에 대한 주석 혹은 관련절에 대한 것이다. 이때 본 시스템에서는 주석의 내용상 중요성 및 크기를 고려하여 짧은 것은 핫 링크로 처리하였다. 또한 중요도가 큰 주석은 하나의 노드로 형성하여 참조 링크로 연결하였다. 또한 각 절과 관련된 절은 하나이상일 경우가 대부분이므로 다중 링크로 처리하였다. 또한 각 장의 서문, 연대기, 기타 사진, 지도등은 각각 하나의 노드로 형성되어 있으며 이들은 지역 링크로 처리되어 있어서 지역 맵을 통해 사용자가 인지할 수 있다. 또한 영상(image) 내의 앵커는 대부분 참조 링크의 기능을 가진다. [그림 8]에서 볼 수 있는 작은 사각형은 다중 링크를 의미하는 것으로 사용자가 이를 선택하면 화면과 같은 작은 윈도우가 나타난다. 이때 윈도우에는 연결되어 있는 앵커들이 나열되어 나타나는데 사용자는 이들을 일별한 후 선택할 수 있다. 또한 이탤릭체로 되어있는 부분을 참조 링크를 나타내는 것으로 이를 선택하면 새로운 윈도우에 참조 내용이 나타난다. 검은 바탕에 화살표를 가진 네모난 표시는 이동 링크로써 현 윈도우를 유지시켜주면서 현재의 내용 및 지역 맵의 내용을 목적 노드의 것으로 바꾸어 준다. [그림 9]는 이미지 링크를 통해 다른 노드의 내용을 참조하고 있는 것을 보여주고



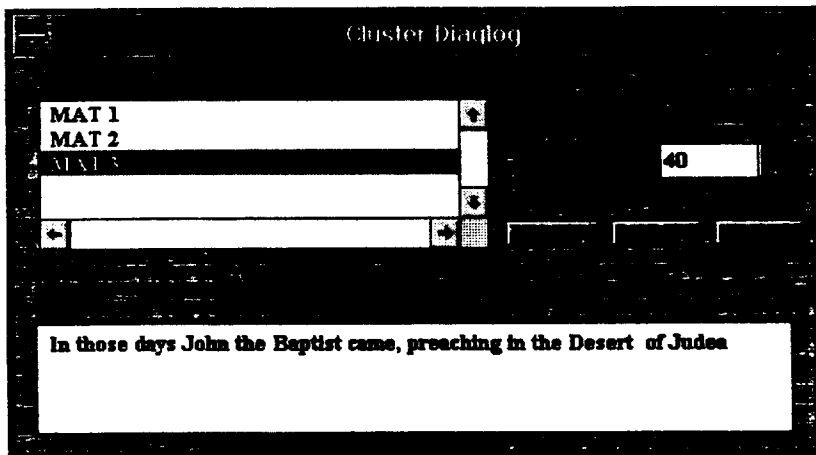
[그림 9] 이미지 링크를 통한 참조



[그림 10] 탐색 윈도우



[그림 11] 지역 맵을 이용한 관련 노드의 참조



[그림 12] 클러스터 윈도우

있다.

사용자는 저자가 연결한 링크를 통해서 계속적인 탐색 항해를 할 수도 있으나 이외에도 클러스터 정보를 이용하여 내용상 관련된 노드를 살펴볼 수도 있다. 성경의 경우 각 클러스터 단위는 장으로 구성되어지는데 앞으로 각 부노드 단위에서도 클러스터를 형성할 수 있도록 확장할 계획이다(그림 12).

성경 시스템을 본 시스템의 응용 분야로 선정하여 구축하는 과정에서 발견된 기능의 미비점이나 문제점은 다음과 같다. 우선 검색을 위한 인덱스 구성이 단순한 전문 색인(full-text indexing)만으로는 부족하다는 것이다. 따라서 인위적으로 구성된 색인을 이용한 검색 기능도 추가되어야 할 것으로 보인다. 또한 클러스터 형성 방식을 노드 단위 뿐만 아니라 부노드 단위에서 구성할 수 있어야 할 것으로 보인다. 링크의 표현방식도 좀더 다양할 필요가 있음을 알 수 있었다. 이외에도 방대한 양의 정보들을 통합하여 관리하기 위한 독자적인 저장 관리 시스템을 구축하는 것도 앞으로의 선결과제이다.

## 6. 결 론

본 연구는 텍스트, 그래픽, 이미지, 목소리/사운드 등 멀티미디어 정보를 지원할 수 있는 하이퍼미디어 시스템을 설계·구현하였다. 하이퍼텍스트 시스템의 문제점으로 지적되고 있는 방향상실 문제를 해결하기 위하여 계층적 뷰, 지역 맵과 탐색을 위한 질의장치를 제공하였고 브라우징을 효율적으로 수행하기 위하여 그래프식 링크와 클러스터링 방식을 통합하여 탐색항해할 수 있도록 하였다. 또한 하이퍼미디어 시스템을 성경 관련정보를 하이퍼 텍스트

화하는데 적용함으로써 본 시스템의 유용성을 보이고 기능과 성능을 검증하였다.

앞으로도 본 시스템을 지속적으로 발전시켜 나가고 새로운 개념을 도입할 계획이다. 현재 본 시스템에서는 노드의 태그된 내용을 작성하여 노드/링크 구성기, 인덱스 구성기 등을 이용하여 실제 링크 및 인덱스를 구성하고 있다. 앞으로는 통합 편집기를 구축하여 대화식으로 이러한 정보를 구축할 계획이다. 또한 탐색기능을 더욱 보강하여 브라우징 장치와 잘 통합되어 이용되도록 할 예정이다.

## 참고문헌

- (1) 정영미, 정보검색론, 구미무역, 1993.
- (2) Conklin J., "Hypertext: An Introduction and Survey", IEEE Computer, pp. 17-41, Sep. 1987.
- (3) Coombs, "Hypertext, Full Text and Automatic Linking", Proceedings of SIGIR'90, 1990.
- (4) Croft W.B. and Turtle H., "A Retrieval Model Incorporating Hypertext Links", Hypertest'89 proceeding, pp. 213-224, Nov. 1989.
- (5) 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.
- (6) Frakes, W. B. and Baeza0yates, R., Information Retrieval, Prentice Hall, 1992.
- (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) H. Vann Dyke Parunak, "Don't Link ME In: Set Based Hypermedia for Taxonomic Reasoning", Hypertext'91 Proceeding, pp. 233-242, Dec. 1991.
- (9) Halasz F. G., "Reflectins On Note cards: Seven Issues For The Next Generation of Hypermedia Systems", Comm. of ACM, Vol. 31, No. 7, pp. 836-851, Jul. 1988.
- (10) J. Nielsen, "Introduction to Hypertext and Hypermedia", SIGGRAPH 1990 Course Notes No. 3, Aug. 1990.
- (11) 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.
- (12) Parsave,K, Chignell,M, Knoshafian, s and Wong, H., Intelligent Database: Object-oriented, Deductive, Hypermedia Technologies, Wiley, 1989.
- (13) Peter A. Gloor, CYBERMAP-Yet Another Way of Navigation in Hyperspace", Hypertext:91 Proceeding, pp. 107-122, Dec. 1991.
- (14) Rodrigo A,B, Ben Shneiderman, "Identifying Aggregates in Hypertext Structure", Hypertext'91 Proceeding, pp. 63-74, Dec. 1991.
- (15) Salton, G., Automatic Text Processing, Addison-Wesley, 1989.
- (16) Shneiderman, B., Designing the User Interface, Second edition, Addison-Wesley, 1992.
- (17) Walter. M, "IRIS Intermedia: Pushing the Boundaries of Hypertext", The Seybold Report on Publishing System, August 1989.