

論文96-33B-3-4

효율적인 링크 형성을 지원하는 멀티미디어 통합편집기의 설계 및 구현

(Design and Implementation of an Integrated Multimedia Editor for Effective Link Creation)

金廷勳 *, 高英坤 **, 崔潤哲 **

(Jung-Hun Kim, Young-Kon Ko, and Yoon-Chul Choy)

요 약

비순차적 정보접근이 가능한 하이퍼미디어 시스템의 구축 시 저작자의 부담을 덜어주기 위하여 링크의 형성 과정이 용이해야한다. 그러나 대부분의 기존 하이퍼미디어 시스템은 하나의 링크를 형성하기 위해 여러번의 번거로운 작업을 거쳐야 하는 단점을 지니고 있다. 또한 텍스트 뿐만아니라 멀티미디어 데이터에 대한 앵커 형성 작업이 쉽지 않다는 단점이 있다. 그러므로 효율적인 하이퍼미디어 시스템 구축을 지원하기 위해서는 손쉽게 링크를 형성할 수 있는 저작환경이 요구된다. 이러한 기존의 하이퍼미디어 시스템의 단점을 보완하기 위해 본 논문에서는 멀티미디어 데이터의 링크 형성을 쉽게 할 수 있는 편집기 도구를 설계하고 구현한다. 링크 형성 과정은 세가지 방법으로 제공되는데 저작자는 편리한 방법을 상황에 따라 선택하여 사용할 수 있다. 또한 멀티미디어 정보를 포함하고 있는 노드의 효율적인 저작을 위해 멀티미디어 데이터를 통합, 처리할 수 있는 환경을 제공한다.

Abstract

To reduce an authors burden in hypermedia system that allows non-sequential information, the process of creating links must be easy. However, most of the conventional hypermedia systems possess two difficulties. First, the author must go through several troublesome process to create a single link. Secondly, it is not easy to create an anchor in text or other multimedia data. Therefore, in order to support effective construction of hypermedia system, the editing environment must provide an easy method to create links. In this paper, to resolve the weaknesses of conventional hypermedia system as mentioned above, an editing tool is developed and implemented to easily create the links of multimedia data. There are three methods in creating links and a user can select a convenient method in given circumstances. And for the efficient production of nodes composed of multimedia information, we provide an authoring environment to integrate and process those informations.

* 正會員, 現代電子 소프트웨어연구소
(Software R&D Center, Hyundai Electronic Co.,)

** 正會員, 延世大學校 컴퓨터科學科 멀티미디어/그래픽스 研究室

(Multimedia/Graphics Lab., Dept. of Computer Science, Yonsei University)

※ 본 논문은 '94년도 한국과학재단의 연구비지원에 의해 수행되었음. (과제번호: 941-0900-069-2)

接受日字: 1995年10月23日, 수정완료일: 1996年2月16日

I. 서 론

1945년 Vannevar Bush에 의해 고안된 하이퍼텍스트(hypertext) 개념은 최근 컴퓨터의 데이터 처리 속도 향상과 대용량 정보의 압축 및 저장이 가능해짐에 따라 문자, 영상, 그래픽, 사운드, 애니메이션, 비디오 등의 멀티미디어 데이터를 포함하는 시스템으로까지 적용되고 있다^{[5][8][10]}.

하이퍼미디어 시스템은 기존의 출판물과는 몇가지 중요한 측면에서 차이점이 있다. 정보의 순차적인 접근을 지원하는 기존의 방법과는 달리 하이퍼미디어 시스템에서는 관련된 내용의 비순차적인 접근을 허용한다. 이는 정보획득의 용이성을 제공한다. 또 접근 매체가 문자에 제한되어 있는 기존의 방법과는 달리 하이퍼미디어 시스템에서는 문자, 영상, 그래픽 등의 정적인 매체 이외에도 사운드, 애니메이션, 비디오 등의 동적인 매체에도 접근할 수 있어, 전달되는 정보의 내용을 쉽게 이해할 수 있는 장점이 있다^[14].

이러한 하이퍼미디어 시스템의 구축을 위해서 요구되는 링크형성 작업이나 멀티미디어 데이터의 통합 처리와 같은 작업은 저작자에게 많은 부담을 주는 작업일 뿐만아니라 일관된 시스템 구성을 방해하는 요인이기도 하다^{[4][16]}. 이에 정보의 비순차적인 접근을 지원하고 멀티미디어 정보의 효율적인 처리를 위한 편집기의 필요성이 대두된다.

비순차적 정보접근이 가능한 하이퍼미디어 시스템의 저작에 대한 저작자의 부담을 덜어주기 위해서는 링크의 형성과정이 용이해야 한다. 그러나 기존 대부분의 하이퍼미디어 시스템은 하나의 링크를 형성하기 위해 여러번의 번거로운 작업을 거쳐야 하는 단점을 지니고 있다. 또한 텍스트 이외의 멀티미디어 데이터에 대한 앵커 형성작업이 쉽지 않다는 단점이 있다. 따라서 하이퍼미디어 시스템 구축의 성공여부는 링크형성의 용이성과 링크의 개념을 어떻게 멀티미디어 데이터로 확장할 수 있는가에 달려있다고 볼 수 있다^[7].

이러한 기존의 하이퍼미디어 시스템의 단점을 보완하기 위해서, 본 논문에서는 멀티미디어 데이터의 링크형성과정의 용이성을 제공할 수 있는 편집기 도구를 설계하고 구현한다. 또한 멀티미디어 정보를 포함하고 있는 노드의 효율적인 제작을 위해 멀티미디어 데이터를 통합, 처리 할 수 있는 환경을 제공한다. 이러한 통합편집기는 구현의 용이성과 확장 가능성을 위해 객체

지향 방식으로 설계되었으며 윈도우즈상에서 객체지향 언어인 C++로 구현하였다^[12].

본 논문의 구성으로서, II장에서는 하이퍼미디어의 기본개념과 통합편집기에서 제공하는 링크형성 방법, 멀티미디어 데이터의 공간 배치에 관해 논하였고 III, IV 장에서는 본 시스템의 설계와 구현, 하이퍼미디어 응용 시스템의 저작과정에 대해 다루었다. 마지막으로 V장에서는 기존의 타 시스템과 비교하였으며 결론을 내렸다.

II. 멀티미디어 통합 편집기의 구성

1. 하이퍼텍스트와 하이퍼미디어

1945년 Bush가 제안한 Memex 이후 1965년 Ted Nelson은 인간의 사고 방식과 유사한 비순차적인 정보접근 방식을 설명하기 위해 하이퍼텍스트라는 단어를 처음으로 사용하였다. 하이퍼텍스트 시스템에서는 관련된 내용을 탐색항해(navigation)나 브라우징(browsing)을 통하여 찾아 볼 수 있게 한다. 이러한 정보 획득 과정은 이미 연결되어 있는 링크를 통해 한 노드로부터 다른 노드까지 여행을 통하여 이루어진다^[7].

하이퍼미디어는 문자이외의 다른 매체도 다룰 수 있는 하이퍼텍스트 개념의 확장이다. 하이퍼미디어 시스템에서의 저작자(author)는 문자, 영상, 그래픽, 사운드, 애니메이션, 비디오 등의 데이터에 링크되어 있는 노드를 생성하여, 하이퍼텍스트 시스템에서와 같이 사용자가 탐색항해나 브라우징을 통하여 원하는 정보를 획득할 수 있게 한다. 하이퍼텍스트 개념을 발전시킨 하이퍼미디어 시스템의 예로서는 OWL사의 Guide 시스템, 브라운대학의 Intermedia, Asymetrics사의 ToolBook, 메릴랜드 대학의 HyperTIES, Apple사의 HyperCard 등이 있다^{[3][7][13][17]}.

하이퍼미디어 시스템을 구성할 때 고려하여야 할 요소중 가장 중요한 요소중의 하나가 링크를 형성하여주는 메카니즘이다. 구성된 각 노드간의 관계를 어떻게 맺어주는가에 따라 하이퍼미디어 시스템의 성격이 달라지기 때문이다^{[10][16]}. 본 논문에서는 노드의 구성 방식과 함께 특히 링크를 좀 더 효율적으로 형성하는 방식에 대해 초점을 맞추었다.

2. 멀티미디어 데이터의 링크 형성과정

본 시스템에서 지원되는 링크들은 모두 단방향 링크

로서, 목적노드의 종류 및 연관 정도에 따라 서로 다른 링크를 지원한다. 간략한 설명을 요구할 때 사용하는 핫링크(hot link), 목적앵커가 또 다른 링크를 가질 수 있는 참조링크(reference link), 시작노드가 사라지고 목적노드가 현재 윈도우에 나타나는 이동링크(move-to link), 여러개의 목적노드를 갖는 다중링크(multiple link)가 본 시스템에서 사용되는 링크이다. 본 시스템은 LDM(Link Direct by Manipulation) 방법을 기본으로 하여 여러 형태의 링크를 형성할 수 있게 하였다.

멀티미디어 데이터에 대한 링크 형성과정은 문자, 영상, 그래픽 등의 정적인 매체의 링크형성과 사운드, 애니메이션, 비디오 등의 동적인 매체의 링크 형성과정으로 분류할 수 있는데, 본 논문에서는 정적인 매체와 사운드 매체에 한정하여 논한다.

1) 텍스트데이터의 링크 형성과정

텍스트정보에 링크를 형성한다는 것은 원래의 텍스트에 태그(tag)를 삽입하는 과정으로 생각해 볼 수 있다. 이때 삽입된 태그는 노드와 링크정보, 그리고 인덱스정보를 생성하기 위하여 분석기를 통해 분석된다. 본 통합편집기에서는 텍스트데이터를 импорт(import)할 경우 원래의 텍스트데이터와 같은 내용의 새로운 화일을 생성하여 그 화일에 태그를 삽입한다. 이렇게 함으로써 원래의 텍스트데이터는 그대로 보존될 수 있다. 본 통합편집기에서 제공하는 텍스트데이터에 태그를 삽입하는 방법을 사용자 인터페이스에 따라 분류하면 다음 3가지로 분류할 수 있다.

· 자판 입력(Form-Fillin) 을 통한 링크형성

저작자는 근원앵커를 선택한 뒤 목적앵커 입력 다이얼로그 박스를 띄운다. [그림 1]에서 보듯이 이 다이얼로그 박스에는 링크 타입을 선택할 수 있는 체크박스가 있으며 목적앵커의 논리적 주소를 입력할 수 있는 부분이 있다. 목적앵커의 주소를 입력하고 OK 버튼을 클릭하면 링크가 형성된다. 통합편집기에서는 저작자에게 목적앵커 주소에 대한 부담을 주지 않기 위해서 시스템이 사용하는 물리적인 주소를 사용하지 않는다. 그러나 논리적인 주소를 사용한다 하더라도 저작자가 목적앵커의 주소를 입력할 때 구문오류(syntactical error)를 범할 가능성이 상당히 크다는 단점을 내포하고 있다. 또한 모든 근원앵커에 대한 목적앵커를 입력해야 하는 번거로움도 존재한다. 그러나 기존의 대부분의 저작도구는 이러한 사용자 인터페이스를

채택하고 있어 작업시간과 저작자의 노력이 많이 요구되는 단점을 지니고 있다.

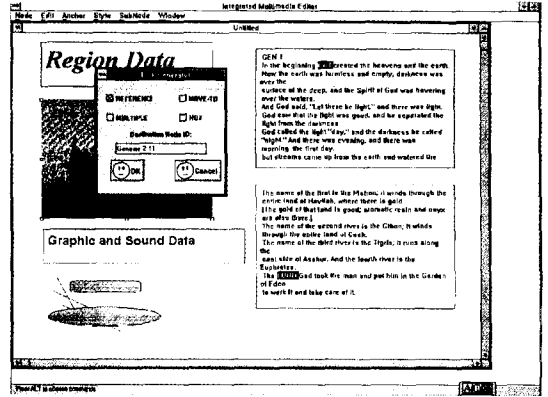


그림 1. 자판을 통해 목적앵커를 입력하는 사용자 인터페이스

Fig. 1. User interface to create destination anchor through keyboard.

· 브라우징 판넬(Browsing Panel)을 이용한 링크 형성

저작자는 앞의 방법과 같이 근원앵커를 선택, 링크 타입을 선택한 뒤에 브라우징 판넬을 통해서 목적지의 논리적인 주소를 찾아가게 된다. 목적앵커 주소를 선택한 뒤에 OK 버튼을 클릭함으로써 링크는 형성된다. 이러한 링크형성 방식은 [그림 2]에서 볼 수 있다.

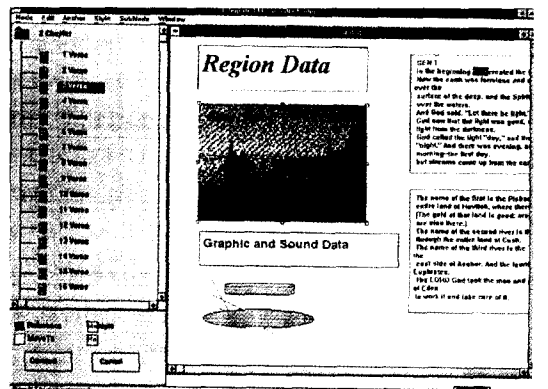


그림 2. 브라우징 판넬을 사용한 사용자 인터페이스

Fig. 2. User interface using browsing panel.

이 방법의 장점은 입력시의 구문오류의 가능성을 배제할 수 있으며 저작자가 목적앵커 주소를 잘 알고 있

어야 할 필요성이 적다는 것이다. 그러나 목적앵커 주소에 대한 계층구조가 깊을 때에는 하나의 링크를 형성하는데 여러 경로를 거쳐야 하는 단점을 내포하고 있다. 또 같은 수준의 목적앵커가 많아서 한 화면에 나타낼 수 없을 때는 스크롤링을 해야하는 번거로움이 존재한다. 기존의 저작시스템 중에는 툴복과 가이드가 이러한 인터페이스를 제공한다.

· 그룹링킹을 이용한 링크형성

앞의 두 방법은 하나의 링크를 생성하기 위해서 목적앵커의 주소를 입력하거나 브라우징을 해야하는 번거로움이 있었다. 다시말해서 여러번의 번거로운 작업을 거쳐야만 하나의 링크를 형성할 수 있다는 단점이 존재한다. 이는 적은 양의 텍스트데이터에 대해서는 유용할지 모르지만 대용량의 데이터에 대해서는 저작자에게 많은 부담을 주게 된다. 본 논문에서 제공하는 세 번째 방법은 다음과 같다.

목적앵커에 대한 정보를 저작자에게 응용제작 초기에 입력 받은 뒤, 목적앵커를 계층 관계에 따라 분류하여 몇개의 리스트 박스를 통해 저작자에게 보여준다. 저작자는 여러개의 근원앵커를 선택, 등록한 뒤 그룹링크 다이얼로그 박스를 띄운다. 이 다이얼로그 박스에는 저작자가 등록된 근원앵커와 목적앵커 주소 리스트가 나열된다. 저작자는 근원앵커와 목적앵커를 선택하고 링크타임을 선택함으로써 링크형성을 완료한다. 즉, 응용제작 초기에 입력 받은 목적앵커 주소를 이용하여, 목적 앵커의 선택과정을 최소화했다는 장점을 가진다. 또한 이 방법에서는 여러개의 근원앵커가 하나의 목적 앵커와 연결 관계를 가질 때, 여러개의 근원앵커와 한 개의 목적앵커를 선택하여 단 한번의 작업으로 링크를 형성할 수 있다. 이는 앞에서 살펴본 링크형성 방법과 비교하여 볼 때 단순하고 빠른 링크형성 방법이 될 수 있으며 구문오류의 가능성을 배제할 수 있다는 장점이 있다. [그림 3]은 이러한 링크형성 예를 보여주고 있다. 이러한 링크들은 [그림 4]와 같이 태그로 삽입된다. 그룹링킹을 이용한 링크형성 방법은 목적 주소가 계층관계를 이루고 있을 때 효율적으로 처리될 수 있다. 본 논문에서는 성경(bible) 하이퍼미디어 시스템을 제작하는 실험을 행하였다. 성경의 구성은 창세기, 출애굽기, 등과 같은 책(book) 단위가 66개 존재하며 각 책 단위의 하위레벨로는 장(chapter)단위가 존재하며 그 하위레벨로는 절(verse)단위가 존재한다. 이러한 특성에 따라 목적주소를 레벨별로 입력하게 되면

적은 노력으로 모든 목적주소의 등록이 가능하다. 근원 앵커를 등록된 목적주소와 연결시켜 줌으로써 별도의 키보드 입력 없이 쉽게 링크를 형성할 수 있다. 이러한 방식은 목적주소의 수가 무수히 많을 경우 위에서 설명한 자판을 통한 입력방식이나 브라우징을 통한 입력 방식의 단점인 반복작업을 피할 수 있다.

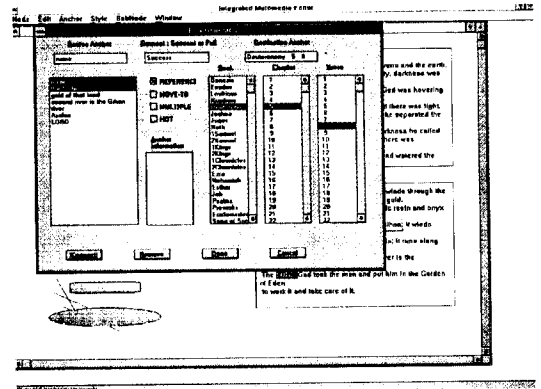


그림 3. 그룹링킹을 이용한 사용자 인터페이스
Fig. 3. User interface using group linking.

2) 영상데이터와 그래픽데이터의 링크형성 과정
영상과 그래픽에서는 하나의 영상객체, 그래픽 객체가 앵커로서 동작할 수 있도록 하였을 뿐 아니라 한 영상데이터 내에 여러개의 영역(region)을 형성하여 각각의 영역이 하나의 앵커로서 동작할 수 있도록 하였다. 이를 위해 영상내에 영역을 구성할 수 있는 영상 편집기를 따로 구현하였는데 이 편집기를 통해 저작자가 마우스를 통하여 다양한 모양의 영역을 생성할 수 있게 하였다. 이렇게 편집된 영상은 (.rgn) 형태의 화일로 저장되며 근원앵커가 된다. 각 근원앵커에 대한 목적지주소는 다이얼로그 박스를 통해서 입력할 수 있다. 물론 그래픽이나 영상의 경우도 텍스트와 마찬가지로 세가지 방법의 인터페이스를 통해 링크를 형성할 수 있다.

A few days later, when (\link ref in 0)~Jesus~Ruth 4(\link moveto in 0)~9~~joshua 6 17~again entered(\link hot in 0)~Capernaum~1Kings 5 9~, the people heard that he had come home. So many gathered that there was no room left, not even out side the door, and he preached the word to them. Some men came, bringing to him a (\link ref in 0)~paralytic~1Samuel 16 6~,carried by four of them.

그림 4. 태그가 삽입된 텍스트데이터
Fig. 4. Text data with tags.

3) 사운드 데이터의 링크 형성과정

복합노드로 저작자가 원하는 사운드 데이터를 임포트한다. 이때 임포트되는 사운드 데이터는 저작자가 선택한 아이콘으로 대표되며 이 아이콘이 사운드 데이터에 대한 앵커의 역할을 담당한다. 사운드 앵커도 위 세 가지 인터페이스를 통해 링크를 형성할 수 있다.

3. 멀티미디어 데이터의 공간배치

문자, 영상, 그래픽, 비디오 등의 멀티미디어 정보는 시간적동기와 공간적동기를 통해 사용자에게 보여줄 수 있다¹⁹⁾. 본 논문에서는 시간적동기는 고려하지 않았으며 공간적동기를 지원할 수 있는 편집기능을 제공한다. 멀티미디어 데이터를 포함한 복합노드의 구성은 다음 [그림 5]와 같다

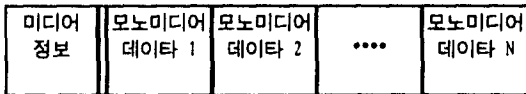


그림 5. 복합노드의 구성

Fig. 5. Internal structure of composite node.

각각의 모노미디어는 자신의 시작위치와 크기 정보인 공간정보를 가지고 있으며 이동이나 크기 변화시에 새로운 공간정보를 생성하게 된다. 본 통합편집기를 이용하면 [그림 6]과 같은 멀티미디어를 포함하고 있는 복합노드를 쉽게 구성할 수 있다.

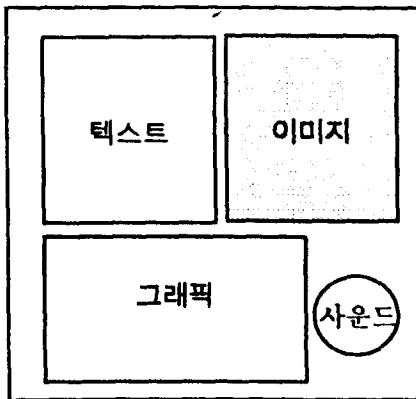


그림 6. 복합노드에서의 미디어 배치

Fig. 6. Media layout in composite node.

통합편집기의 주요 기능은 각각의 미디어편집기에서 생성된 데이터들을 복합노드(composite node)로 임포트하여 한 노드내에 통합시키는 일이다. 임포트된 각

데이터들은 부노드를 형성하는데 이 부노드는 하나의 객체로 인식되어 크기를 자유롭게 조절할 수 있으며 이동이 가능하다. 여기서 부노드는 문자영역, 영상영역, 그래픽영역, 사운드영역 등을 의미한다. 통합편집기에서는 문자정보, 영상정보, 그래픽정보, 사운드정보들을 동일한 정보단위로 생각하여 서로 관련된 내용을 링크로 연결할 수 있다.

4. 미디어편집기

본 논문에서는 통합편집기 이외에도 각각의 미디어를 독립적으로 다룰 수 있는 문자편집기, 영상편집기, 그래픽편집기, 사운드편집기 등의 편집기를 제공하며 이들 편집기는 한 데스크탑에서 동시에 작업할 수 있는 특징이 있다.

문자편집기는 문자 입력과 링크를 형성하기 위한 도구이다. 영상편집기의 주요 기능은 근원앵커로 사용될 영상의 영역을 형성하는 일이다. 영역형성은 마우스 클릭을 통해서 이루어지고 형성된 영역정보는 원래 영상의 데이터 뒷 부분에 첨가되어 응용에서 이 영역정보를 이용할 수 있도록 한다. 그래픽편집기는 2차원 그래픽객체를 형성하고 그래픽영역을 형성한다. 이렇게 생성된 데이터는 복합노드로 임포트되어 원하는 위치로 이동될 수 있다. 사운드편집기에서는 사운드의 녹음과 재생, 오려두기, 복사, 붙이기 기능과 에코우, 음량 조절 등의 기능이 제공된다. 애니메이션과 비디오 부분은 차후 연구과제로서 확장 가능성만 제공된다.

III. 시스템의 설계 및 구현

1. 기본개념과 메세지전달 그래프

멀티미디어 통합편집기의 설계는 Rebecca의 설계방식을 따른다¹⁵⁾. Rebecca의 설계방식은 시스템에서 사용될 클래스들을 설정한 뒤 각 클래스들이 수행해야 할 행동(behavior)을 정의한다. 관련있는 클래스들의 집합을 부시스템이라고 하며 전체 시스템은 몇개의 부시스템으로 구성된다. 부시스템이 정의되면 클래스 상호간, 클래스와 부시스템, 부시스템 상호간에 이루어지는 메세지 전달과정을 나타내는 메세지전달 그래프를 작성한다.

전체적인 부시스템간의 메세지전달 그래프는 [그림 7]과 같다. 여기서 부시스템은 둥근 사각형(rounded rectangle)로 표시하였으며 클래스는 사각형(rectangle)으로 표시하였다.

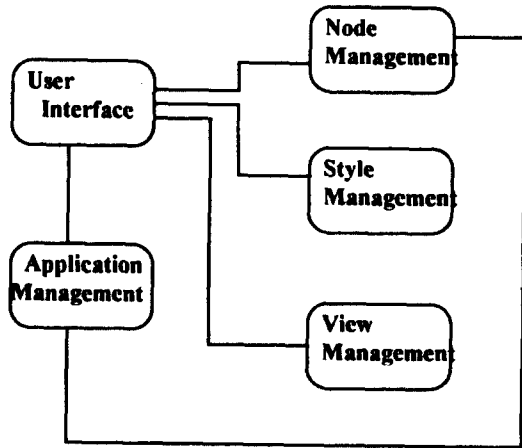


그림 7. 전체적인 메세지전달 그래프
Fig. 7. Overall graph of message delivery.

사용자 인터페이스 부시스템(User Interface Subsystem)은 사용자가 발생시키는 모든 이벤트를 처리한다. 이벤트에는 메뉴 선택, 키보드 입력, 마우스 클릭, 이동, 드래그 등이 포함된다.

응용관리 부시스템 (Application Management Subsystem)은 응용시스템 수준에서의 기능을 수행한다. 새로운 응용의 생성과 개발, 저장의 기능을 수행한다. 응용을 처음 생성할 때 저작자는 관련정보를 정의한다. 이 정보에는 응용에서 사용되는 데이터의 저장 위치, 각 노드들의 정보, 새로운 데이터가 생성되거나 임포트될 때 시스템 생성 화일이름, 각 종류의 링크 모양 등이 포함되어있다.

또 이 부시스템에서는 여러명의 저작자가 제작한 부응용을 하나의 응용으로 통합하는 과정이 수행된다. 통합된 응용은 태그가 분석되는 과정과 정보의 효율적인 접근을 위해서 클러스터링 과정을 거친다.

노드관리 부시스템(Node Management Subsystem)은 노드의 생성, 개발, 저장과 같은 노드에 관련된 전반적인 내용을 다룬다. 또한 노드관리 부시스템에서는 한 노드를 구성하는 문자, 영상, 그래픽, 사운드 등의 데이터를 관리하는 기능이 수행된다. 형식관리 부시스템(Style Management Subsystem)은 노드의 형식과 부노드의 형식을 정의한다. 뷰관리 부시스템(View Management Subsystem)은 노드의 내용을 사용자에게 보여 주는 역할을 한다. [그림 8]은 이러한 각 시스템별 구성도를 보여주고 있다.

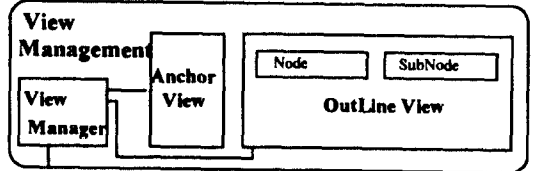
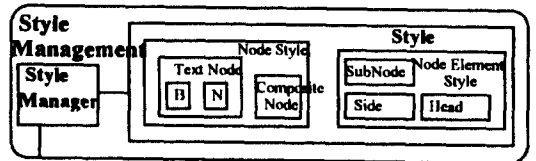
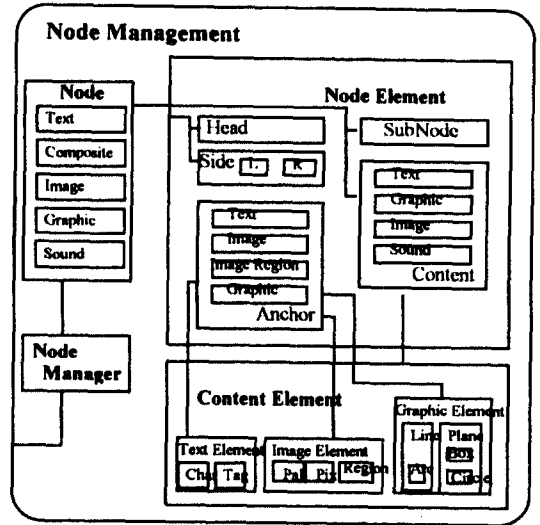


그림 8. 노드관리, 형식관리, 뷰관리 부시스템의 상세 설계
Fig. 8. Design diagram of node, style, and view sub-system.

2. 복합노드의 저장구조

복합노드에서 각각의 미디어는 동일한 형태로 저장된다. 이는 편집이 완료된 후 행하는 분석작업을 용이하게 하고자 함이다. 복합노드의 저장구조는 [그림 9]와 같다.

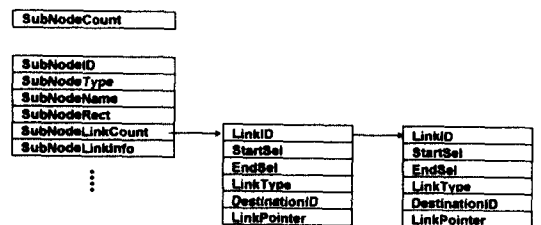


그림 9. 복합노드 및 링크정보의 저장구조
Fig. 9. Storage structure of Composite node and link.

[그림 9]는 각각의 모노미디어들이 모여 한 복합 노드를 구성하는 예를 보여주고 있는데 각각의 미디어, 즉 복합노드에서의 부노드는 수에 제한없이 링크정보를 가질 수 있다. 이러한 링크정보는 LINKINFO 라는 구조에 의해 저장된다.

3. 사용자 인터페이스

본 통합편집기에서는 윈도우상에서 메뉴를 이용한 대화방식의 사용자 인터페이스를 제공하며 아래와 같은 메뉴로 구성되어 있다.

통합편집기와 문자편집기에서는 마이크로소프트 윈도우즈 응용에서 제공하는 개방, 저장, 오퍼두기, 복사, 붙이기 기능을 지원한다. 통합편집기에서 필요로 하는 값은 다이얼로그 박스를 통해서 저작자에 의해 직접 정의되거나 이미 정의된 디폴트 값을 이용한다.

통합편집기의 전체적인 사용자 인터페이스는 풀다운 메뉴(pull down menu)방식과 단축키(shortcut key)를 지원하고 부분적으로 직접적인 조작이 가능한 인터페이스로 구현된다¹⁶⁾.

텍스트 부노드는 마우스 버튼을 누른채로 이동하여 원하는 크기가 되면 마우스 버튼을 놓음으로써 생성된다. 이렇게 생성된 텍스트 부노드에서 저작자는 문자를 입력하거나 импорт할 수 있다.

IV. 하이퍼미디어 응용시스템의 제작과정

1. 개요

멀티미디어 통합편집기로 작성된 데이터는 본 연구실에서 개발한 HyperWIS(Hypermedia With Intelligent Search)에 적용되어 실제로 하이퍼미디어 응용을 제작할 수 있는 환경을 제공한다^{11) 12) 18)}. HyperWIS 시스템은 Dexter 모델을 기본으로 하고 있으며 탐색과 브라우징을 통합하여 원하는 정보를 쉽게 획득할 수 있는 특징이 있다. 또 본 통합편집기를 통해 생성된 데이터의 링크정보를 분석하는 기능, 정보들의 연관성에 따라 클러스터를 생성할 수 있는 기능이 있어 통합편집기와 더불어 완전한 하이퍼미디어 시스템을 구성한다. 통합편집기와 HyperWIS 간의 상호연관성은 [그림 10]과 같다.

2. 응용 제작

응용으로 성경을 선택하였는데 이는 성경이 방대한 양의 정보와 정보 상호간의 연관관계가 많은데에 기인

한다.

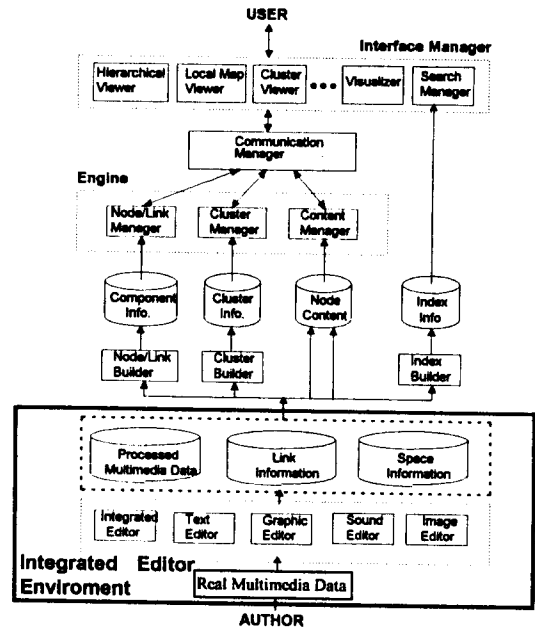


그림 10. 통합편집기와 HyperWIS의 구성
Fig. 10. Architecture of integrated editor and Hyper WIS.

문자는 아스키(ASCII) 코드값을 импорт하거나 문자 편집기에서 직접 입력한 데이터를 사용한다. 성경 응용은 크게 구약과 신약으로 나누어지며 구약, 신약은 책(book)으로 구성되어 있다. 책은 다시 총 2887개의 장(chapter)으로 나누어지는데 장 단위가 한 노드가 된다. 한 노드내에 삽입되어야 할 태그 정보는 앞에서 설명한 링크 생성과정을 거쳐 원래의 문자정보에 삽입된다.

영상편집기에서 처리된 영상은 필요한 부분에 사각형 또는 다각형 영역이 형성되어 응용에서 앵커로 사용된다. 여기서 생성되는 앵커들은 계층구조를 가질 수 있다. 그래픽편집기에서 작성된 그래픽 객체는 영상데이터와 함께 새로운 객체를 형성하거나 그래픽 자체만으로 하나의 객체를 형성시켜 응용에서 사용된다. 이상의 절차를 [그림 11]과 같이 나타낼 수 있다.

3. 탐색항해와 브라우징

사용자는 HyperWIS의 계층적 뷰(hierarchy view)를 통해 정보획득의 시작점을 찾을 수 있다. 정보 획득의 시작점을 얻는 다른 방법은 검색을 통해 시작점을 얻는 것이다. 시작점을 찾은 사용자는 그 시작점과 밀

접한 관련이 있는 노드가 지역 맵(local map)에 나타나도록 하여, 원한다면 관련이 있는 노드로 이동할 수 있다.

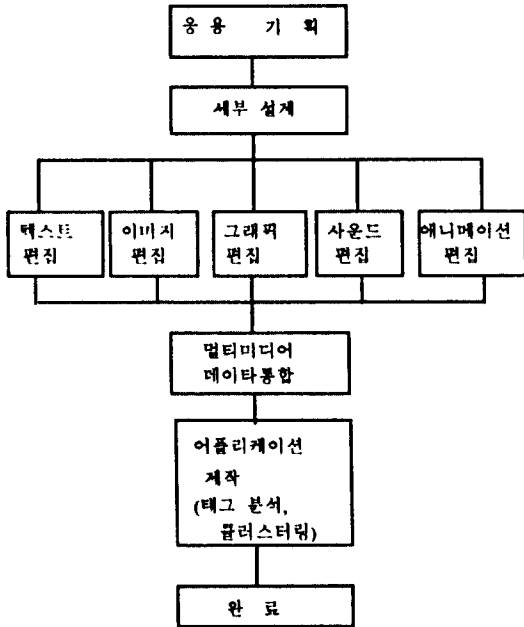


그림 11. 응용제작 과정
Fig. 11. Application building process.

지역 맵에 있는 내용은 노드간의 유사도(similarity level)를 계산하여 얻은 클러스터 정보에 근거하여 사용자에게 보여지게 된다^[11]. 이러한 클러스터 뷰(cluster view)를 통하여 정보를 획득하게 함으로써 사용자가 탐색중에 겪게되는 방향상실(disorientation)의 위험을 어느정도 배제한다. [그림 12]는 HyperWIS의 한 작업환경을 보여주는 그림이다.

V. 타 시스템과의 비교 및 결론

본 논문에서 제공하는 링크형성 방법과 기존의 툴북(ToolBook)이나 가이드(Guide)와 같은 하이퍼미디어 시스템에서의 링크형성 방법을 비교해보면 다음과 같다.

툴북에서는 앞에서 살펴본 바와 같이 LDM 방법과 스크립트언어를 사용하는 방법, 2가지를 제공한다. LDM 방법은 여러 페이지(툴북에서는 한 정보 단위가 페이지가 된다) 중 원하는 목적앵커를 찾아가서 선택하는 방법이고, 사용하기는 어려우나 스크립트언어를

사용하면 링크에 다양한 기능을 추가할 수 있다. 또 구문오류를 검사하기 위해 디버깅을 할 수 있는 기능이 제공되지만 하나의 링크를 형성하기 위해 여러번의 작업을 거쳐야 하는 단점이 있다^[3].

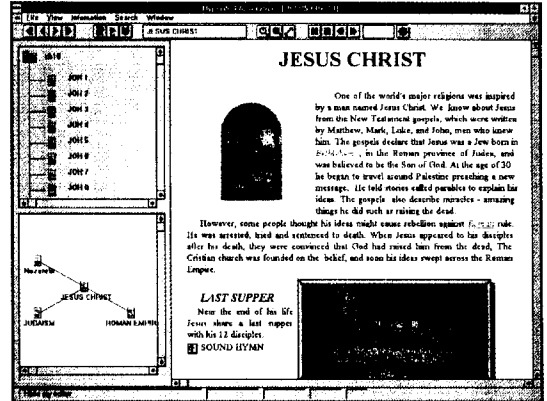


그림 12. HyperWIS의 작업 환경
Fig. 12. User environment of Hyper WIS.

가이드에서는 근원앵커와 목적앵커 간의 관계에 따라 여러 가지 다양한 형태의 링크를 지원하며 툴북과는 조금 다른 방법의 링크 형성과정을 제공한다. 링크를 형성하는 과정에서 선택된 임의의 근원앵커에 대한 목적앵커의 주소가 바로 전단계의 목적앵커와 같다면, 따로 링크를 형성하지 않아도 현재의 근원앵커에 대한 목적앵커의 주소가 자동적으로 바로 전의 목적앵커의 주소와 같게 된다. 그러나 바로 전의 근원앵커에 대한 목적앵커의 주소와 현재 선택된 근원앵커에 대한 목적앵커에 대한 주소가 서로 다를 경우 툴북에서와 마찬가지로 여러번의 조작을 거쳐야 링크가 형성된다^[7].

본 논문의 통합편집기에서는 이러한 번거로운 링크 형성 과정을 최소한의 조작을 거쳐 형성할 수 있도록 설계하고 구현하였다. 링크형성 과정은 사용자 인터페이스 측면에서 3가지 다른 방법을 제안하고 저작자가 편리한 방법을 선택, 사용할 수 있도록 하였다. 첫째는 목적앵커의 주소를 직접 입력하는 방법이며, 둘째는 브라우징 패널을 이용하여 목적앵커를 선택하는 방법이다. 세번째 방법은 목적주소가 계층관계를 이루고 있을 때 목적앵커들을 계층구조에 따라 몇개의 리스트 박스를 통해 나타내고 저작자는 원하는 목적앵커를 마우스로 선택할 수 있도록 한 방법이다. 첫번째 방법의 단점은 구문오류가 발생될 수 있고 번거로운 입력 작업이 따른다는 점이고 두번째 방법은 구문오류의 가능성을

배제하였으나 목적앵커에 대한 주소를 브라우저해야 하는 번거로운점이 있다. 즉, 통합편집기에서 제공하는 두가지 방법은 틀복이나 가이드처럼 하나의 링크를 형성하기 위해 여러번의 조작을 거쳐야 한다. 이는 적은 양의 데이터에 대해서, 또 링크 관계가 적은 경우에는 훌륭한 방법이 될 수 있다. 세째 방법은 목적앵커의 주소를 계층 관계에 따라 초기에 한번만 등록하도록 하여 저작시에는 근원앵커와 목적앵커 간의 관계를 연결하도록 한 방법이다. 앞서서도 언급한 바와 같이 목적 주소의 수가 무수히 많을 경우 위에서 설명한 자판을 통한 입력방식이나 브라우저를 통한 입력방식의 단점인 반복작업을 피할 수 있다. 이는 기존의 방법과는 아주 다른 특성을 제공하는 것이다. 즉 목적주소의 갯수와 무관한 방법으로, 링크 생성시 가능한 저작자에게 적은 노력으로 링크를 생성시킬 수 있도록 한 방법이다.

멀티미디어 데이터의 통합은 각 편집기에서 생성된 데이터들을 통합편집기로 임포트하여 서로간에 통합, 처리함으로써 이루어진다. 특히 본 시스템은 문자에 대한 링크 뿐만아니라 멀티미디어 데이터에도 링크를 형성할 수 있는 환경을 제공한다. 영상과 그래픽에서는 영역을 형성하여 앵커를 나타내며 사운드에서는 그 사운드를 대표하는 아이콘으로 앵커를 나타낼 수 있도록 하였다. 이들 앵커들의 목적주소는 다이얼로그 박스를 통해 입력할 수 있다. 현재 본 시스템을 이용한 응용은 성경 이외에도 백과사전을 대상으로 하이퍼미디어 응용시스템을 구축하고 있다.

참 고 문 헌

- [1] 고영곤, 최윤철 "효율적인 탐색과 브라우저를 지원하는 하이퍼미디어 시스템의 사용자 인터페이스 설계", *대한인간공학회지*, 제12권, 제1호, 1993.6
- [2] 고영곤, 최윤철, "멀티미디어 정보의 효율적인 검색을 위한 하이퍼미디어 시스템의 설계와 구현", *한국통신학회 논문지*, 제18권, 제8호, 1993.8
- [3] Asymetrix, *Using ToolBook*, 1989-1991
- [4] Ben Shneiderman, "Direct Manipulation: A Step Beyond Programming Languages", *Computer*, june, 1983.
- [5] Ben Shneiderman and Greg Kearsley, *Hypertext Hands-On!*, Addison-Wesley Pub., 1989.
- [6] Ben shneiderman, *Designing the User Interface*, Addison-Wesley Publishing Company 1992.
- [7] Conklin, J., "Hypertext: An Introduction and Survey", *IEEE Computer*, Vol2 No9, Sept. 1987.
- [8] Emily B., Joseph D., *Hypertext/Hypermedia Handbook*, McGraw-Hill Publishing Company, Inc. 1991.
- [9] Hodges, M.E., Sasnett, R.M. "A Construction Set for Multimedia Application," *IEEE Software*, january, 1989.
- [10] Jakob Nielsen, *Hypertext and Hypermedia*, ACADEMIC PRESS, INC. 1990.
- [11] Kamran P., Mark C., Setrag K., Harry W., *Intelligent Databases*, WILEY, 1989
- [12] Ori Gurewich, Nathan Gurewich, *Visual C++ Programming*, SAMS 1994.
- [13] Microsoft, *Multimedia Authoring and tools Guide*, Microsoft Press. 1991.
- [14] Michael Bieber, Issues in Modeling a Dynamic Hypertext Interface, *Proc. of Hypertext91*, 1991.
- [15] Rebecca Wirfs-Brian Wilkerson, Laurea Wiener, *Designing Object-Oriented Software*, Prentice Hall, 1990.
- [16] Thomas C.Rearick, *Automating the Conversion of Text Into Hypertext*, Big-Sience Company, 1991.
- [17] Walter, M., *IRIS Intermedia: Pushing the Boundaries of Hypertext*, The Seybold Report on Publishing Systems, 1989.
- [18] Y.K. Ko, S.H. Jung, Y.C. Choy, Design and Implementation of an Integrated Information Space for Hypermedia System, *Proc. of ITI95*, 1995.

— 저 자 소 개 —



金 廷 勳(正會員)

1968년 5월 26일생. 1991년 2월 서울시립대학교 전산통계학과 졸업(이학사). 1994년 2월 연세대학교 컴퓨터과학과 졸업(이학석사). 1994년 ~ 현재 현대전자(주) 소프트웨어 연구소 주임연구원.

관심분야는 하이퍼미디어, 멀티미디어 데이터처리



高 英 坤(正會員)

1967년 2월 4일생. 1991년 2월 한국과학기술대학 컴퓨터과 졸업(공학사). 1993년 8월 연세대학교 컴퓨터과학과 졸업(이학석사). 1993년 8월 ~ 현재 연세대학교 컴퓨터과학과 박사과정.

관심분야는 멀티미디어 및 하이퍼미디어, 문서처리



崔 潤 哲(正會員)

1950년 6월 20일생. 1973년 2월 서울대학교 전자공학과 졸업(공학사). 1975년 6월 Univ. of Pittsburgh (공학석사). 1979년 6월 Univ. of California, Berkeley, Dept. of IE&OR(공학

박사). 1979년 8월 ~ 1982년 7월 Lockheed사 및 Rockwell International사 책임연구원. 1982년 9월 ~ 1984년 1월 Univ. of Washington, 전산학과 박사과정. 1990년 9월 ~ 1992년 1월 Univ. of Maisachusetts. 연구교수. 1984년 3월 ~ 현재 연세대학교 컴퓨터과학과 교수. 관심분야는 멀티미디어 및 하이퍼미디어, 지리정보시스템(GIS)