

# 폼에 기반한 HTML 문서 생성 방법

최 준 용<sup>†</sup> · 김 병 기<sup>†\*</sup>

## 요 약

대규모 웹사이트의 정보구조는 계층적 형태로 구성되는 것이 일반적이며, 이런 구조에서 형제 노드들은 동일하거나 유사한 태그와 내용으로 이루어진다. 본 논문은 형제노드에서 공통된 부분의 집합이라는 의미로 "폼"이라는 용어를 사용하고, 각 노드의 폼과 내용을 분리하여 설계하는 방법을 제안하며, 폼과 내용으로부터 HTML 문서를 생성하는 알고리즘을 제시한다. 이 방법은 폼의 재사용, 유지보수 용이성, 저작 비용의 감소 등의 이점을 얻을 수 있다.

## A Method of Form-Based HTML Documents Generation

Jun-Yong Choi<sup>†</sup> · Byung-Ki Kim<sup>†\*</sup>

### ABSTRACT

An information structure of large size hypermedia application is usually hierarchical, and the sibling nodes in this structure have same or similar tags and contents. A word "form" that means the common set of sibling nodes in the hierarchical information structure is used in this paper. It proposes a design method that divides form and content from nodes and it proposes HTML page generation algorithm from forms and contents. This method has reusability of form, maintainability of documents and reduction of cost for authoring.

### 1. 서 론

WWW은 인터넷상에서 간단한 하이퍼미디어 링크에 의해 분산된 정보를 일반적이고 효율적으로 볼 수 있으며, 사용법이 편리하고 멀티미디어 자료를 손쉽게 제공받을 수 있어서 사용자들의 수가 폭발적으로 늘어가고 있다[1]. 또한 WWW은 응용범위가 매우 넓어 하루에도 많은 양의 하이퍼미디어 어플리케이션이 세계 각국에서 저작되고 있다.

WWW은 클라이언트/서버 모델을 이용하고 있으며, 서버에서 클라이언트로 전달하는 문서는 HTML을 사용하여 작성된다[2][3]. WWW 환경에서 어플리케이션

을 설계하고 개발하는 것은 기존의 소프트웨어 개발과 비교할 때 환경 자체가 트리므로 개발하는 방법도 서로 일치하지 않는다. WWW 어플리케이션은 개발자 외에도 저작자, 사서, 음악가, 그래픽 설계가 등이 필요하며, 미적, 인지적인 측면은 기존의 소프트웨어 공학 환경과 다른 매우 중요한 부분이다. 따라서 대규모 하이퍼미디어 어플리케이션을 개발하는 것은 상당히 복잡하며, 이를 위한 특별한 방법론과 도구가 필요하다[3][4].

한편, 하이퍼미디어 어플리케이션의 개발 시 적은 비용으로 양질의 산출물을 얻을 수 있도록 하기 위하여 저작자는 적절한 방법론을 선택해야 하며, 이와 관련한 방법론은 소프트웨어공학으로부터 유추될 수 있다[5]. 이제 하이퍼미디어 어플리케이션은 소규모의 시스템을 수작업으로 개발하는 것에서 벗어나 특정 개발

<sup>†</sup> 정 회 원 : 전남대학교 대학원 전산통계학과  
<sup>†\*</sup> 종 신 회 원 : 전남대학교 전산학과 교수  
논문접수 : 1998년 7월 28일, 심사완료 : 1998년 11월 6일

방법이나 도구를 사용함으로써 양질의 대규모 시스템을 개발할 수 있도록 바뀌어지고 있다.

하이퍼미디어 어플리케이션을 개발하기 위한 과정은 방법에 따라 다소 차이가 있으나 명세화(Specification), 설계(Design), 내용 생성(Contents generation), 저작(Authoring), 출판, 생산 및 분배(Production and/or Distribution) 등과 같은 행위를 유발하며[6], 설계는 시스템의 구조, 사용성(Usability), 유지보수 등과 관련한 매우 중요한 단계이다[5][7].

본 논문은 대규모의 멀티미디어 자료를 구축한 후 홍보 또는 상품 쇼핑 목적으로 웹사이트(Web-site)를 개발하는 회사나 조직체 등과 같이 자료구조가 계층적 형태로 구성되어 있는 대규모 하이퍼미디어 어플리케이션을 저작할 때 저작 및 유지보수의 용이성을 증대할 수 있는 설계 방법을 제시한다. 이러한 하이퍼미디어 어플리케이션의 정보구조에서 형제 노드(노드는 하나의 HTML 문서를 의미한다)들은 동일하거나 유사한 형태의 태그를 공통적으로 사용한다. 또한 각 노드의 내용(contents)은 서로 유사한 종류의 정보로 구성된다[8][9].

따라서 형제 노드간에 공유되어야 할 부분을 설계 단계에서 정의하여 DB에 저장하여 두고 공통되지 않은 부분만 따로 작성한다면, HTML 편집기 등의 방법으로 전체 페이지를 저작하는 것에 비하여 상대적으로 적은 시간과 노력으로 하이퍼미디어 어플리케이션을 개발할 수 있다.

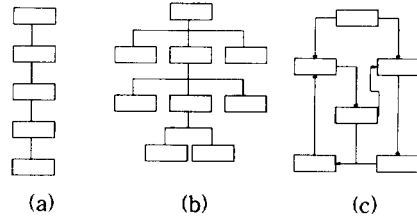
본 논문의 구성은 다음과 같다. 2장은 하이퍼미디어 설계에서 관심이 되는 정보 구조와 설계동향에 관하여 서술한다. 3장은 본 논문이 제시하는 폼과 내용을 분리하여 HTML 문서를 설계하는 방법에 관하여 서술한다. 4장은 이 논문의 결론을 서술한다.

## 2. 관련연구

### 2.1 하이퍼미디어 어플리케이션의 정보 구조

하이퍼미디어 어플리케이션의 저작 과정에서 이슈가 되는 것 중의 하나는 정보의 구조이다. 정보들은 명시적이고, 서로 융화되어야 하며, 조작이 용이하여야 하는데 이것은 정보의 내부구조에 의존적이다. 제공해야 할 정보의 내용이나 목적, 정보의 제공 방법 등에 따라 선형구조, 계층적 구조, 그래프(네트워크)구조 등의 형태를 취할 수 있으며, 이들이 조합된 형태의 구조

도 가능하다. 그러나, 많은 하이퍼미디어 어플리케이션은 계층적 구조를 사용한다[6].



(그림 1) 하이퍼미디어 어플리케이션의 정보 구조  
(a) 선형 (b) 계층적 (c) 그래프

(Fig. 1) Information structure of hypermedia application  
(a) Linear (b) Hierarchical (c) Graph

### 2.2 하이퍼미디어 어플리케이션의 설계 동향

하이퍼미디어 어플리케이션에서 설계와 관련된 많은 연구자들의 논문이 제시되었다. 이 논문들 중 많은 것들은 웹 문서의 공동 저작[10][11], HTML 구문검사를 고려한 저작환경[3], 대규모 문서의 높은 생산성[12], 객체지향 기법 적용[13], 하이퍼미디어 저작도구[4][14] 등과 같이 특정 목적을 대상으로 연구되었다.

또한 HDM[15], OOHDM[16], RMM[17] 등은 초기의 하이퍼미디어/하이퍼텍스트에 관한 설계 방법이나 특별한 형태의 하이퍼미디어 개발 문제나 데이터를 근원에 둔 모델에서의 구현 등에 관해서는 언급하지 않고 있다. DB 설계 방법에 근원을 둔 데이터 중심적 설계를 사용하는 방법은[18] 유지보수 문제를 용이하게 하지만 사용성 문제 해결에는 적합하지 않다. WSDM[7]은 사용성, 유지보수 용이성 등의 문제를 해결하려고 시도하고 있지만 사용자를 등급별로 분류하여야 하는 어려운 점이 있다.

템플릿(template) 또는 폼(form)에 대하여 언급하고 있는 논문들로는 WebWriter[14], 최용준[19], W3DT[5]와 같은 논문들이 있다. 그러나 WebWriter는 설계 단계에서부터 폼을 고려하는 것이 아니라 필요할 경우(노드간에 공통되는 부분이 있을 때) 폼을 만들어서 사용한다는 개념이며, 최용준, W3DT 등은 형제 노드들 중의 특정 노드에 대하여 이미 만들어진 폼에 없는 새로운 형태를 추가하기 어려운 점이 있다.

### 2.3 저작 방법의 비교

지금까지 개발된 하이퍼미디어 저작은 서로 다른

관점의 방법론에서 제시되었으므로 일관적인 비교는 쉽지 않다. 본 절에서는 각 방법의 특징과 단점을 기술하는 수준에서 본 논문의 저작 방법을 고찰한다. 제안된 방법과 다른 방법의 비교를 <표 1>에 정리하였다[14][20][21][22] [19].

<표 1> 각 방법들의 비교  
<Table 1> Comparison of each methods

	특징	단점
Amaya [21]	<ul style="list-style-type: none"> <li>Amaya 웹 브라우저에서 직접 문서 편집</li> <li>HTML에 대한 지식 불필요</li> </ul>	<ul style="list-style-type: none"> <li>대규모 저작 어려움</li> </ul>
WebWriter [14]	<ul style="list-style-type: none"> <li>일반 웹 브라우저에서 직접 문서 편집</li> <li>HTML에 대한 지식 불필요</li> </ul>	<ul style="list-style-type: none"> <li>편집 절차 번거로움</li> <li>대규모 저작 어려움</li> </ul>
W3Objects [20]	<ul style="list-style-type: none"> <li>일반 웹 브라우저에서 직접 문서 편집</li> </ul>	<ul style="list-style-type: none"> <li>HTML에 대한 지식 필요</li> <li>서버 시스템 구조 복잡/복수함</li> <li>대규모 저작 어려움</li> </ul>
[22]	<ul style="list-style-type: none"> <li>동적인 HTML 문서 작성 가능</li> <li>시스템 관리자의 문서 작성 용이</li> </ul>	<ul style="list-style-type: none"> <li>시스템의 부하 증대</li> <li>대규모 저작 어려움</li> </ul>
[19]	<ul style="list-style-type: none"> <li>체계적인 홈페이지 관리</li> <li>관리자의 계층화</li> </ul>	<ul style="list-style-type: none"> <li>관리 전용 브라우저 필요</li> </ul>
제안 방법	<ul style="list-style-type: none"> <li>품의 계사용</li> <li>품과 내용에 의한 HTML 문서 자동 생성</li> <li>대규모 문서 저작의 어려움이 상대적으로 감소됨</li> </ul>	<ul style="list-style-type: none"> <li>HTML에 대한 지식 필요</li> </ul>

### 3. 품에 기반한 하이퍼미디어 설계

#### 3.1 정보 구조

대규모의 멀티미디어 자료를 구축한 후 홍보 또는 상품 쇼핑 목적으로 웹사이트(Web-site)를 개발하는 회사나 조직체 등의 정보구조는 계층적 형태로 구성되는 것이 일반적이며, 대부분의 사회 환경이 계층적으로 구성되었다는 점을 고려한다면 계층적 구조는 인간의 의식 구조와 일치한다. 또한 하이퍼미디어를 계층적으로 구성하였을 때 사용자의 사용성을 증가시킬 수 있으며, 특별한 장치를 사용하지 않더라도 방향상실의 위험을 감소시킨다.

본 논문은 이와 같이 계층적 구조에 적합한 대규모 하이퍼미디어의 설계를 중점적으로 고려하고 있다. (그림 1)의 정보구조에 나타난 예지는 상하노드간에 링크 관계가 있는 것으로 생각할 수 있으며, 계층적 구조에서 형제 노드들은 동일하거나 유사한 형태의 태그를 공통적으로 사용하며, 각 노드의 내용 또한 서로 유사한 종류의 정보로 구성된다[8][9].

따라서 노드간에 공유되는 부분을 설계 단계에서 미리 정의하여 DB에 저장하여 두고 공통되지 않은 부분만 작성한다면, 기존의 방법으로 전체 하이퍼미디어 문서를 저작하는 것에 비하여 상대적으로 적은 시간과 노력으로 HTML 문서를 생성할 수 있다.

#### 3.2 품과 내용의 분리

HTML 문서는 태그와 내용의 집합으로 이루어진다. 태그는 속성과 값으로 된 쌍의 집합을 가질 수 있으며, 내용은 몇 개의 단에서부터 몇 페이지 정도의 긴 문단으로 이루어진 텍스트이거나 DB에 파일로 저장되어 있는 그림, 소리, 영상 등으로 이루어진다.

아래의 그림은 멀티미디어 농수산물 주문판매 시스템[23] 중 녹차 품목에서 발췌한 쌍계 작설차와 화개 작설차의 쇼핑을 위한 HTML 문서이다.

```

<html>
<head>
<title>농수산물 주문/쇼핑</title>
</head>

<body bgcolor="white">
<P> <H2> 농산물/녹차/화개작설차 </H2>
<P> 화개 장터 전통 아생 녹차입니다
<IMG SRC = "/agr/image/speaker.gif" WIDTH=30 HEIGHT=30>
<EMBED SRC = "/agn/ga271.au" AUTOSTART=TRUE>
<P>
<IMG SRC="/agr/html/image/ga151.jpg" WIDTH=500 ALIGN= CENTER>
시스템에 조연이 있으면 <A HREF="mailto:ychoi@ superse.chonnam.ac.kr"> 전자우편 </A>으로 연락바랍니다.
</body>

</html>
    
```

(a) 화개 작설차

```

<html>
<head>
<title>농수산물 주문/쇼핑</title>
</head>

<body bgcolor="white">
<P> <H2> 농산물/녹차/쌍계작설차 </H2>
<P> 천년의 향기-차의 고향 지리산 화개동의 아생차를 10대 토박이가 장작불 가마솥과 백탄 솥불에서 만든 녹차입니다
<IMG SRC = "/agr/image/speaker.gif" WIDTH=30 HEIGHT=30>
<EMBED SRC = "/agn/ga272.au" AUTOSTART = TRUE>
<P>
<IMG SRC="/agr/html/image/ga151.jpg" WIDTH=500 ALIGN= CENTER>
시스템에 조연이 있으면 <A HREF="mailto:ychoi@ superse.chonnam.ac.kr"> 전자우편 </A>으로 연락바랍니다.
</body>

</html>
    
```

(b) 쌍계 작설차

(그림 2) 녹차 쇼핑을 위한 HTML 문서의 예  
(Fig. 2) Example of HTML pages for green tea shopping  
(a) Hwagea green tea (b) Sanggye green tea

(그림 2)의 (a)와 (b)를 살펴보면 두 문서간의 태그는 동일하며, 각 태그의 동일한 위치에서의 내용은 서

로 같은 종류-예를 들면, 같은 형태의 텍스트이거나, 그림 파일-로 구성되어 있는 것을 볼 수 있다. 이것은 "농산물/녹차"에 해당하는 모든 상품이 동일한 형태로 구성되어 있다는 것을 의미한다. 이와 같이 형제 노드들 간에 동일한 부분의 집합을 본 논문에서는 폼이라는 용어로 사용하며, 이 폼은 설계 단계에서 정의되어야 한다는 것을 의미한다. 이런 폼들을 DB화하여 저장하여 둔다면, 같은 폼을 사용하는 동일한 노드들의 문서는 내용만을 따로 분리하여 기술할 수 있다.

```

<html>
<head>
<title>농수산물 주문/쇼핑</title>
</head>

<body bgcolor="white">
<P>
<H2> &Content1 </H2>
<P> &Content2
<IMG SRC = "/agri/image/speaker.gif" WIDTH=30 HEIGHT=30>
<IMG SRC = &Content3 WIDTH=500 ALIGN = CENTER>
<EMBED SRC = &Content4 AUTOSTART=TRUE>
시스템에 조연이 있으시면 <A HREF="mailto:yjchoi@superse.
chonnam.ac.kr"> 전자우편 </A>으로 연락바랍니다.</body>
</html>
    
```

(그림 3) 녹차 쇼핑을 위한 폼  
(Fig. 3) The form for green tea shopping

(그림 3)은 (그림 2)의 (a), (b)에서 공통된 부분을 추출한 폼이다. 본 논문은 각 노드별로 폼 안에 들어가야 할 내용을 기술하기 위하여 프로그래밍 언어에서 사용하는 변수의 개념을 도입하였다. 이 변수는 각 노드의 실제 내용을 기술하기 위하여 사용되며, 하나의 변수 이름(variable name)은 동일한 폼을 사용하는 형제노드에서 각각 서로 다른 내용의 값을 갖는다. 본 연구에서는 변수가 &Content와 숫자의 조합으로 이루어진다고 가정한다. &는 영어의 Content와 변수 Content를 구분하기 위하여 편의상 사용하였다.

```

&Content1 = 농산물/녹차/화계작설차
&Content2 = 화계 장터 전통 야생 녹차입니다
&Content3 = "/agri/html/image/ga151.jpg"
&Content4 = "/agri/ga271.au"
    
```

(a) 화계 작설차

```

&Content1 = 농산물/녹차/쌍계작설차
&Content2 = 천년의 향기-차의 고향 지리산 등의 야생차를 10대
토박이가 장작불 가마솥과 백탄 솥불에서 만든 녹
차입니다
&Content3 = "/agri/html/image/ga152.jpg"
&Content4 = "/agri/ga272.au"
    
```

(b) 쌍계 작설차

(그림 4) 녹차 쇼핑을 위한 내용  
(Fig. 4) The contents for green tea shopping  
(a) Hwagea green tea (b) Sanggye green tea

(그림 4)는 (그림 2)의 (a), (b)에서 공통되지 않은 내용만을 기술한 것이다. (그림 3, 4)와 같은 저작방법은 (그림 2)의 저작 방법에 비해 상대적으로 적은 노력과 시간이 소요된다. 또한 폼은 재사용이 가능하며, 한번의 수정으로써 같은 폼을 사용하는 모든 노드들에게 동일한 수정 효과를 주므로 하이퍼미디어 어플리케이션의 유지보수를 용이하게 한다.

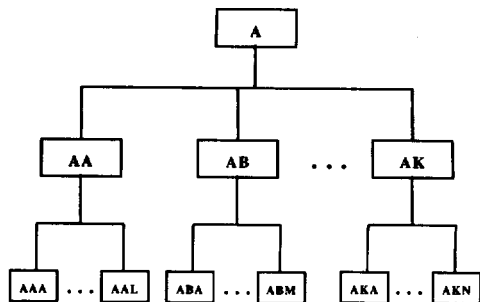
작은 규모의 하이퍼미디어 어플리케이션에서는 공통된 부분이 많지 않으므로 본 논문의 저작 방법이 적절치 않을 수 있으나, 대규모의 하이퍼미디어 어플리케이션은 형제 노드간에 공통된 부분이 매우 많을 것으로 생각된다. 실제로 농수산물 주문판매 시스템에서는 동일한 형태의 페이지가 한 종류의 상품에 대하여 수~수십 개 정도이다.

### 3.3 폼과 내용의 표현 방법

#### 1) 노드와 키의 사상

대규모 하이퍼미디어 어플리케이션은 많은 수의 HTML 문서로 이루어진다. 본 논문에서 제시한 방법으로 하이퍼미디어 어플리케이션을 저작하기 위하여 설계 단계에서 많은 수의 폼을 정의하여야 하며, 각 노드에 해당하는 내용 파일을 별도로 저작하여야 한다. 따라서 하나의 폼 파일과 여러 개의 내용 파일을 서로 연관시키는 방법과 각각의 내용 파일들을 구분할 수 있는 방법이 필요하다.

본 논문에서는 각 노드의 ID를 키(key)의 개념으로 사용한다. 키는 정보구조의 각 계층별로 영문자나 숫자 또는 이들의 조합으로 구성될 수 있으며, 하위 계층의 키는 상위 계층의 키를 물려받고 추가로 한자리 또는 두 자리의 숫자나 영문자가 첨가됨으로써 (그림 5)와 같은 계층적 구조를 이루게 된다[8].



(그림 5) 키의 계층적 구조  
(Fig. 5) The hierarchical structure of key

위의 방법에 의하여 노드는 ID 대신 하나의 유일한 키를 가지고, 품은 형제 노드들에 대하여 하나의 품이 존재하므로 상위 노드의 키를 품의 ID로 사용할 수 있다. 이 경우 루트 노드의 품은 없으며, 실제로 루트 노드는 단 하나 존재하므로 품이 필요하지 않다.

품과 내용을 저장하는 방법은 구현하는 방법에 따라 다양할 수 있다. 예를 들면, 첫째, 한 파일에는 모든 품을, 다른 파일에는 모든 내용을 저장한다. 둘째, 한 파일에 하나의 품과 관련된 여러 개의 내용을 저장한다. 셋째, 모든 품과 내용을 각각 하나의 파일에 저장한다. 넷째, DBMS를 이용한다. 그 외에도 계층적으로 관리하는 방법 등 다양할 수 있으나 본 논문에서는 이의 구현에 대한 연구는 고려치 않았다.

2) 공통 내용 추출의 극대화

품은 저작자의 의도에 따라 공통 내용을 최대한 추출하여 기술할 수 있다. 예를 들면, (그림 4)의 &content1의 경우 "농산물/녹차/"가 공통적으로 쓰인다. 이 경우 (그림 6)과 같이 품 파일에서 "<H2> 농산물/녹차/&Content1 </H2>"로 기술하고, 내용 파일에서 각각 "&content1 = 화개 작설차", "&content1 = 쌍계 작설차"와 같이 기술할 수 있다. 공통 내용을 최대한 추출하여 품으로 정의함으로써 내용을 저작할 때 드는 노력은 비례적으로 감소한다.

```
<H2> 농산물/녹차/&Content1 </H2>
```

(a) (그림 3)을 변형한 품

```
&Content1 = 화개작설차
```

(b) 화개 작설차

```
&Content1 = 쌍계작설차
```

(c) 쌍계 작설차

(그림 6) 녹차 쇼핑을 위한 품과 내용

(Fig. 6) The form and contents for green tea shopping

2) 내용 부분의 태그 허용

(그림 4)의 (a)에서 "&Content2 = 화개 장터 전통 야생 녹차입니다" 부분은 (그림 7)과 같이 "&Content2 = <B>화개 장터</B> 전통 야생 녹차입니다"로 기술할 수 있다. 이와 같이 노드간에 서로 유사한 내용이기는 하나 완전히 일치하지 않아서 노드의 내용을 약간 변경하려고 할 경우 변수의 값으로 내용 부분에 HTML 태그를 사용할 수 있다.

또한 내용의 값으로 태그를 사용할 수 있으므로 품에서 정의되지 않았으나 특정 노드에만 필요한 내용을

삽입함으로써 같은 형제 노드와는 다른 형태의 HTML 문서를 생성할 수 있다.

```
&Content2 = <B>화개 장터</B> 전통 야생 녹차입니다
```

(그림 7) 내용 부분의 태그  
(Fig. 7) Tag in a content

3) 텍스트 파일 이름 허용

HTML 문서 내에는 몇 개의 문단으로 이루어진 긴 텍스트도 존재할 수 있다. 이와 같이 긴 텍스트는 내용 파일 안에 기술할 수도 있지만, 이런 경우 텍스트 내용을 파일로 저장하고 내용 파일에는 파일의 이름만 기술하는 것이 내용 파일의 복잡성을 감소하는 방법이 될 수 있다. 이를 위하여 (그림 8)과 같이 &Contenti = "텍스트 파일 이름"과 같은 형식을 지원한다. 이 경우 텍스트 파일 이름의 확장자는 txt로 정의하기로 한다.

```
&Content2 = content.txt
```

(그림 8) 내용 부분의 텍스트 파일 이름  
(Fig. 8) Text file name in a content

4) 동일 변수의 중복 사용

(그림 9)와 같이 HTML 문서 내에 동일한 내용을 삽입하려고 할 경우 (그림 10)의 (a)처럼 동일 변수를 중복 사용함으로써 문서 저작의 노력을 감소시킬 수 있다.

```
<HR align = CENTER width = 80%>
<IMG SRC = "/agri/image/speaker.gif" >
<HR align = CENTER width = 80%>
```

(그림 9) HTML 문서의 한 부분  
(Fig. 9) A part of HTML page

```
&Content1 <IMG SRC = &Content2 > &Content1
```

(a) 품

```
&Content1 = <HR align = CENTER width = 80%>
&Content2 = "/agri/image/speaker.gif"
```

(b) 내용

(그림 10) 변수를 중복 사용한 예  
(Fig. 10) Several use of a same variable  
(a) A form (b) A contents

3.4 HTML 문서의 생성

위에서 정의한 방법으로 하이퍼미디어를 저작할 경우, 각 노드별 HTML 문서를 자동으로 생성하여주는 도구를 사용하는 것이 필요하며, 이는 컴퓨터의 도움

을 받는 저작방법을 의미한다. 각 노드별 HTML 문서는 폼 파일의 변수 부분에 지정된 내용을 삽입함으로써 생성된다. 또한 이 알고리즘을 모든 노드에 반복 적용함으로써 웹사이트 전체의 HTML문서를 동시에 생성한다. 입력은 생성할 문서의 키만이 요구되며, 키는 상속되므로 부모 노드의 키는 폼의 키와 동일하다.

HTML 문서를 생성하는 알고리즘은 (그림 10)과 같다. 폼 파일에서 스트링을 하나씩 읽고 그 스트링이 파일 이름인 경우 파일의 내용을 출력하고, 스트링이 변수인 경우 내용 파일에서 동일한 변수의 내용을 출력하며, 그 외에는 스트링을 그대로 출력한다.

```

generateHypermedia()
( For (모든 노드의 KEY)
  generateHTML(KEY)
)

generateHTML(char * KEY)
{ while(!EOF)
  {
  form file에서 string을 1개 읽는다
  if(string == 변수)
  { if(변수 == 텍스트 파일) 텍스트 파일의 내용 출력
    else 변수의 내용 출력
  }
  else string 출력
  }
}
    
```

(그림 11) HTML 문서 생성 알고리즘

(Fig. 11) The algorithm of HTML page generation

#### 4. 결 론

하이퍼미디어 어플리케이션의 정보구조는 제공해야 할 정보의 내용이나 목적, 정보의 제공 방법 등에 따라 선형구조, 계층적 구조, 그래프(네트워크)구조 등의 형태를 취할 수 있으며, 이들이 조합된 형태의 구조도 가능하다. 그러나, 많은 하이퍼미디어 어플리케이션은 계층적 구조를 사용한다.

계층적 구조에서 형제 노드들은 동일하거나 유사한 형태의 태그를 공통적으로 사용하는 경우가 많으며, 각 노드의 내용 또한 서로 유사한 종류의 정보로 구성된다.

본 논문은 설계 단계에서 형제 노드들에 대한 폼을 정의하고, 각 노드에서 공통되지 않은 부분만 별도로 작성하는 방법을 제안하였다.

또한 폼과 내용 파일로부터 HTML 문서를 생성하

는 알고리즘을 제시하였으며, 하나의 폼 파일과 여러 개의 내용 파일을 서로 연관시키고 각각의 내용 파일들을 구분하기 위하여 각 노드의 ID로서 키를 사용하였다.

본 논문의 설계 방법은 폼의 재사용, 동일 변수의 중복 사용, 공통적인 내용 추출의 최대화 측면에서 저작 시간과 노력을 감소시킬 수 있으며, 하나의 폼을 수정함으로써 같은 폼을 사용하는 모든 노드들의 형태를 동시에 수정할 수 있으므로 유지보수의 용이성을 추구할 수 있다. 또한, HTML 문서 내에 몇 개의 문단으로 이루어진 긴 텍스트는 그의 내용을 파일로 저장하고 내용 파일에는 파일의 이름만 기술하는 방법을 사용하여 내용 파일의 복잡성을 감소할 수 있으며, 내용 파일에 태그를 허용함으로써 형제 노드들 중의 특정 노드에 대하여 이미 만들어진 폼에 없는 새로운 형태를 추가할 수 있다.

#### 참 고 문 헌

- [1] Kaj G., Niels O.B., Lennert S., "Designing Dexter-based hypermedia services for the WWW," ACM, Inc. 1997.
- [2] Bayan Pfaffenberger, "World Wide Web Bible," Henry Holt and Co., Inc., 1995.
- [3] Vincent Q., Cecile R., Irene V., "A Structured Authoring Environment for the World-Wide Web," Third international WWW conference, 1995. 4.
- [4] Alica D., Tomas I., Vanesa M., Gabriel G., "RMC: A Tool to Design WWW Applications," Forth International WWW Conference, Vol.1, 1996. 12.
- [5] Martin B., Stefan N., "Developing Structured WWW-Sites with W3DT," WebNet 96 Conference, 1996. 10.
- [6] Athula G., David B.L., and Hohn R., "Hypermedia Authoring," IEEE MultiMedia, Winter 1995.
- [7] O.M.F. De Troyer, C.J. Leune, "WSDM : A User Centered Design Method for Web Sites," Seventh international WWW conference, 1998. 4.
- [8] 최준용, 김병기, "CGI를 이용한 계층적 구조의 웹 문서 생성기에 관한 연구", 춘계학술발표대회 논

문집, 한국정보처리학회, 1998. 4.

[9] 최준용, 유영록, 서성채, 유진호, 장종표, 김병기, "웹 문서 생성기를 이용한 상품 쇼핑 시스템", 제 11회 산학연 멀티미디어 산업기술 학술대회, 한국정보처리학회 멀티미디어 시스템 연구회, 1998. 6.

[10] Dominique D., Vincent Q., Manuel R.S., "Structured Cooperative Authoring on the World-Wide Web," Forth International WWW Conference, Vol.1, 1996. 12.

[11] Jean Paoli, "Cooperative Work on the Network : Edit the WWW!," Third International WWW Conference, 1995. 4.

[12] Marcus Kessler, "A Schema-Based Approach to HTML Authoring," Forth International WWW Conference, Vol.1, 1996. 12.

[13] T.L. Nguyen, X. Wu, S. Sajeev, "Object-oriented modeling of multimedia documents," Seventh International WWW Conference, 1998. 8.

[14] Arturo C., Eric A.B., "WebWriter : A browser-based Editor for constructing Web application," Fifth international WWW conference, 1996. 5.

[15] F. Garzotto, P. Paolini, D. Schwabe, "HDM : a model-based approach to hypertext application design," ACM Transaction on Information Systems, 11(1), 1993.

[16] D. Schwabe, G. Rossi, "The object-oriented hypermedia design model," Communications of the ACM, 38(8), 1995.

[17] T. Isacowitz, E.A. Stohr, P. Balasubramanian, "RMM : a methodology for structured hypermedia design," Communications of the ACM, 38(8), 1995.

[18] Maristella A., Massimo M., Fabio. C., "Automatic authoring and construction of hypermedia for information retrieval," Multimedia Systems, Springer-Verlag, 1995.

[19] 최용준, 임경수, 황도삼, 김종근, "계층적 정보구조의 Web 시스템 관리 기술", 정보처리 논문지, 제5권 제5호, 한국정보처리학회, 1998. 5.

[20] David I, Mark L, Steve C, Santosh S., "W3Objects : Bringing object-oriented technology to the Web," Forth international WWW conference, 1996. 12.

[21] Iree V., Vincent Q., Daniel V., Jose K., Ramzi G., "Amaya-the W3C testbed client," W3C, 1997, <http://www.w3c.org/Amaya/>.

[22] 김현중, 이기호, "HTML을 지원하는 라이브러리를 이용한 웹 문서 생성 시스템의 설계 및 구현", 한국정보과학회논문지(C), 제3권 제4호, 한국정보과학회, 1997. 8.

[23] 김병기, "멀티미디어 농수산물 주문판매 시스템", 최종연구보고서, 1997. 6.

### 최준용



e-mail : jychoi@superse.chonnam.ac.kr  
 1984년 경북대학교 통계학과(학사)  
 1991년 전남대학교 대학원 전산통계학과(이학석사)  
 1995~현재 전남대학교 대학원 전산통계학과 박사과정

관심분야 : 소프트웨어공학, 하이퍼미디어 저작

### 김병기



e-mail : bgkim@chonam.chonnam.ac.kr  
 1978년 전남대학교 수학교육과(학사)  
 1980년 전남대학교 대학원 수학과(이학석사)  
 1981년~현재 전남대학교 전산학과 교수

관심분야 : 소프트웨어공학, 신경망컴퓨터, 초고속정보통신