유지보수에 특화된 웹 문서 작성기의 설계 및 구현*

조영석** · 권용호** · 도재수**

요 약

웹 사이트들은 정보의 양이 방대해지고 웹 문서들의 잦은 생성, 삭제와 수정이 반복되면서 더욱 복잡한 구조의 양상을 띠게 되었으며 사용자가 원하는 최적의 정보에 접근하는 방법 또한 예측하기 힘든 구조를 가지게 되었다. 또한 웹 사이트를 처음 만드는데 들이는 노력에 비해 유지·보수에는 요구되는 비용 때문에 적은 노력을 들이고 있다.

이런 환경에서 웹 문서간의 관계와 그 관계들의 유효성을 분석하여 그 정보가 제공된다면 개발자 뿐 아니라 관리자들도 효과적이고 효율적인 서비스를 제공할 수 있다. 웹 사이트 전체의 구조를 쉽게 파악할 수 있고 하이퍼링크의 유효성을 확인하기 위해 웹 문서의 분석을 통해 하이퍼링크의 구조를 추출하고 체계적인 방법으로 웹 사이트를 관리 하는데 필요한 여러 가지 정보를 제공하여야 한다.

본 논문에서는 HTML 태그를 분석하여 하이퍼링크 정보를 추출, 문서간의 관계를 체계적으로 구조화 하고 관계를 이루는 하이퍼링크가 유효한지 여부를 판단하여 알려주는 두 가지 방법을 사용하여 웹 사이트의 유지·보수를 지원함과 동시에 새로운 문서를 생성·편집할 수 있는 웹 문서 작성 방안을 제안한다.

Design and Implementation of a WebEditor Specialized for Web-Site Maintenance

Youngsuk Cho** · Yongho Kwon** · Jaesu Do**

ABSTRACT

Users of World Wide Web (Web) experience difficulties in the retrieval of pertinent information due to the increased information provided by Web sites and the complex structure of Web documents that are continuously created, deleted, restructured, and updated. Web providers' efforts to maintain their sites are tend to be less than that of site creation due to the expenses required for maintenance.

If information of relationship among Web documents and their validity is provided to Web managers as well as Web developers, they can better serve users. In order to grasp the whole structure of a Web site and to verify the validity of hyperlinks, traversal and analysis of hyperlinks in a Web document are required to provide information for effective and efficient creation and maintenance of the Web.

In this paper, we introduce a Web Editor specialized for Web maintenance. We emphasized on two aspects: first, the analysis of HTML Tags to extract hyperlink information and second, establishment of the relationship among hyperlinked documents, and verification of the validity of them.

Key words : Web Documents Maintenance, Hyperlink Validity, Web Documents Structure

---

* 이 논문은 2005년도 동국대학교 연구년 지원에 의하여 이루어졌음.

** 동국대학교 컴퓨터·멀티미디어학과

1. 서 론

기존의 다양한 정보 서비스 시스템들은 웹을 기반으로 통합되거나 웹 사이트로 구축되어 질 높은 정보의 공유와 보다 안정성 있는 정보의 흐름을 보여주고 있다[1]. 그러나 웹 사이트들은 시간이 지남에 따라, 정보의 양이 방대해지고 웹 문서들의 잦은 생성, 삭제와 수정이 반복되면서 더욱 복잡한 구조의 양상을 띠게 되었으며 사용자가 원하는 최적의 정보에 접근하는 방법 또한 예측하기 힘든 구조를 가지게 되었다.

각종 웹 문서 작성기들은 웹 사이트를 보다 손쉽게 구축할 수 있도록 하고 있다. 하지만 근래의 웹 문서 작성기들은 문서를 작성하는 것에 주로 초점이 맞추어져 있어, 시간이 지남에 따라 복잡한 구조를 가지게 되는 웹 사이트를 효과적으로 유지·보수하는 기능은 다소 취약하다. 따라서 본 논문에서는 웹 문서의 작성과 유지·보수 측면에서 기존 웹 문서 작성기의 문제점을 진단하고 분석하여, 기 작성된 웹 문서들의 링크를 분석하고 하이퍼링크의 유효성을 검증하여 웹 관리자가 효율적으로 시스템을 구축하고 유지·보수하게 하여 보다 나은 품질의 서비스를 제공할 수 있도록 한다.

2. 문제점 분석 및 관련 연구

본 장에서는 기존 웹 문서 작성기의 문제점을 진단, 분석하며, 작성된 웹 문서를 분석하기 위해 필요한 관련 분야에 대해 기술한다.

2.1 현 웹 문서 작성기들의 문제점

기존의 웹 문서 작성기들은 다음과 같은 문제점을 가지고 있다.

① 웹 문서의 유지·보수의 지원이 취약하다.

② 하이퍼링크 되어 있는 웹 문서들의 유효성을 보장하지 못한다.

③ 임의의 웹 사이트의 구조를 한눈에 파악할 수 없어 연결구조의 균형을 유지하는 작업을 지원하지 못한다.

따라서 본 논문에서는 웹 문서의 작성 뿐 아니라 위의 세 가지 취약점을 보완하는 웹 문서 작성기를 제안한다.

2.2 웹 마이닝

웹 사이트의 유지·보수를 지원하기 위해서는 웹 마이닝 기술이 사용된다. 웹 마이닝은 웹으로부터 얻어지는 방대한 양의 정보를 필터링하여 필요한 정보를 찾아내어 이를 분석한다[2, 3]. 웹에서 얻어지는 정보는 로그 데이터와 사용자 프로파일, 컨텐츠, 웹 문서, 하이퍼링크 등이 있으며, 이를 데이터베이스화하여 정보를 자동으로 검색하고 추출한다. 웹 마이닝 기법은 웹 내용 마이닝(Web Content Mining), 웹 사용 마이닝(Web-Usage Mining)과 웹 구조 마이닝(Web Structure Mining)으로 구분된다[4].

2.3 웹 사이트의 구조화

본 논문에서는 웹 사이트를 구조화시키는 방법 중의 하나로 웹 문서를 방향그래프의 자료구조 형태로 생성하고 이것을 트리뷰로 표현하는 방법을 제안한다. 실제 웹 문서의 구조가 방향 그래프로 표현될 경우, 웹 문서 순회에 용이하다. 순회 패턴 탐사를 표현하기 위해서 웹 문서의 태그를 분석하여 하이퍼링크를 추출하고 웹 문서 구조를 트리뷰로 표현한다. 그러나 이 경우, 플래시와 애플릿에 포함된 하이퍼링크 경로는 추출할 수 없어 완전한 구조를 생성하지 못한다. 이러한 히든 링크는 웹 서버의 접근 로그를 조사하여 클릭스트림을 분석하면 추출이 가능하나, 웹 서버의 접근 로그는 웹 서버의 일정 부분을 호스팅 받는 일반적인 사용자의 경우에는 열람 권한이 없으므로, 본 논문에서는

HTML 태그의 하이퍼링크 정보를 추출하여 이를 방향 그래프 형태로 구성하고 트리뷰를 통해 표현하고자 한다.

3. 웹 문서의 유지 · 보수

웹 문서를 유지 · 보수하는데 필요한 자동수집 가능한 정보는 두 가지가 있다. 첫째, 웹 문서의 작성, 수정일자와 둘째, 각 사이트의 홈페이지에서부터의 하이퍼링크를 추적한 문서의 구조이다. 각 웹 문서의 작성 또는 수정일자는 문서의 유효성과 직접적 관계가 없고, 웹 관리자의 결정에 의존적이다. 따라서 본 논문에서는 두 번째 정보를 이용하여 유지 · 보수를 지원한다. 즉, 웹 문서 내의 하이퍼링크를 모두 추출하고 분석하여 SPV(Structured Page Viewer)와 SBL(Search for Broken Link)을 작성한다.

3.1 구조화된 웹 문서 뷰어 - SPV

SPV는 웹의 유지 · 보수 작업 시 웹 사이트 내의 웹 문서들이 어떤 구조로 연결되어 있는가를 한눈에 볼 수 있게 한다. 이는 한 웹 사이트 내의 하이퍼링크 깊이의 균형 정도를 조정하여 웹 사이트 내의 문서들이 균형 잡힌 n-ary 트리 형태에 가까운 그래프를 이룰 수 있게 한다. 완전한 균형을 이루기는 현실적으로 어렵지만 균형 잡힌 트리 형태에 가까울수록 사용자들의 검색 시간이 줄어든다. 즉, 웹 사이트 내에 n개의 페이지가 있으며 모든 페이지에 h개의 하이퍼링크가 있을 경우 사용자의 검색 시간은 다음과 같다.

① 모든 페이지가 선형으로 링크된 경우 : 0(n)
② 한 문서 내에 모든 링크를 포함할 경우
: 0(1) 또는 0(k)
③ 균형 잡힌 n-ary 트리의 경우(순환 링크 제외)
: $(\log_h n)-1 \approx 0(\log_h n)$

①의 경우, 한 페이지에 단 하나의 링크가 있어 깊이가 깊어지고, ②의 경우에는 모든 하이퍼링크가 한 페이지에 있어 h가 증가할 경우, 매우 복잡한 문서가 되고(n * h 링크) 도메인이나 카테고리에 따른 분류에 한계가 있으므로 ③의 경우가 바람직하다.

웹 문서의 태그를 분석하여 하이퍼링크를 추출하고, 추출된 하이퍼링크를 너비 우선 탐색 알고리즘을 적용하여 적합한 자료구조 형태인 방향 그래프로 표현한다. 깊이 우선 탐색 알고리즘을 적용할 때, 웹 페이지가 타 웹 사이트와 하이퍼링크 된 경우, 또는 순환 하이퍼링크를 구성하고 있을 경우 순회가 무한히 계속될 수 있고, 이를 방지하기 위해서는 모든 페이지들에 대해 이미 탐색이 이루어진 페이지와 비교를 해야 한다. 또한 관리자가 유지 · 보수의 단계를 결정할 경우 너비 우선 탐색 알고리즘이 효과적이다.

3.1.1 HTML 태그의 하이퍼링크 분류

(그림 3.1)은 웹 문서의 일부이며 HTML 태그로 구성되어 있다. (그림 3.1)과 (그림 3.2)는 서로 연결된 2개의 HTML 문서의 예이다. 웹 문서의 HTML 태그를 분석하면 해당 문서의 연결 정보들을 얻어올 수 있다. 본 논문에서는 태그들 중, 하이퍼링크를 표현하는 태그만을 처리하기 위해 HTML에 기술되는 태그 속성 중 <a>태그의 href와 <frame>의 src, <form>의 action 속성의 하이퍼링크를 사용한다.

```
                                        …
<form action = “popup.html” method = “get”>
                                        …
<frame src = “top.html” name = “top”>
                                        …
<a href = “link1.html”>링크 #1</a>
<a href = “link2.html”><img src = “image/image1.jpg”></a>
<a href = “http://dongguk.ac.kr”>링크 #2</a>
<a href = “http://w3c.org”><img src = “image/image2.jpg”></a>
                                        …
<iframe src = “bottom.html” name = “bottom”>
                                        …
```

(그림 3.1) HTML 소스코드의 일부분(main.html)

<표 3.1>의 내용 중 형식이 텍스트인 것은 또 다른 문서 링크들을 포함할 수 있기 때문에 이를 추출하여 너비 우선 탐색 알고리즘에 적용한다. 탐색 시 추출하고자 하는 하이퍼링크는 해당 웹 서버에 존재하는 문서들을 대상으로 하며 이 문서들은 해당 웹 서버 자체적으로 보유하고 있는 문서와 웹 서버 외부에 존재하는 문서 두 가지로 구분된다. 이는 파일과 도메인으로 구분되며 외부 하이퍼링크의 경우, 그 하위의 문서정보는 추출할 필요가 없다.

(그림 3.2)는 (그림 3.1)의 웹 문서에서 링크된 link1.html문서로 SPV 기능과 SBL 기능의 예로 사용된다. <표 3.1>는 (그림 3.1)과 (그림 3.2)의 웹 문서에서 추출한 하이퍼링크 이다.

```
...
<a href = "link3.html">링크 #1</a>
<a href = "http://donguk.ac.kr">
<a href = "http://w3q.org"><img src = "image/image3.jpg"></a>
...
```

(그림 3.2) HTML 소스코드의 일부분(link1.html)

〈표 3.1〉 문서의 구조화를 위해 추출한 하이퍼링크

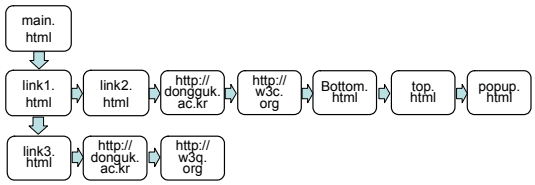
추출된 페이지	태그	속성	하이퍼링크 경로
main.html	<a>	href	link1.html
	<a>	href	link2.html
	<a>	href	http://dongguk.ac.kr
	<a>	href	http://w3c.org
	<iframe>	src	bottom.html
	<frame>	src	top.html
	<form>	action	popup.html
link1.html	<a>	href	link3.html
	<a>	href	http://donguk.ac.kr
	<a>	href	http://w3q.org

3.1.2 웹 문서 순환탐색

(그림 3.3)은 웹 문서 순환탐색 알고리즘으로 특정 웹 문서를 시작 페이지로 하여 하이퍼링크만을 분석하여 연결된 모든 문서에 대해 모든 경로를 단계별로 탐색한다. (그림 3.3)의 알고리즘에서 사용된 AddVisitedDoc()는 문서 d에의 방문을 추가하는 함수이고, isExternal()은 해당 링크가 URL인지 파일인지 판단하는 함수이며, AddLink()는 두 개의 문서를 연결하는 함수이고, isVisited()는 하이퍼링크 h의 방문 여부를 알아내는 함수이다.

```
// 입력 : 웹 사이트의 임의의 웹 문서 d
// 출력 : AddLink()를 통해 d에서 연결된 하이퍼링크 h를
   추출한 방향 그래프
void Find_Hyperlink(document d) {
   // 방문한 문서 d에 대해 방문 추가
   AddVisitedDoc(d) ;
   // 문서 d에 있는 각 하이퍼링크 h
   for each hyperlink h in document d {
      if (isExternal(h) == true) { // 외부 문서(URL)인가?
         AddLink(d, h) ;              // d에서 h로의 연결 생성
      }
      else { // 내부 문서인 경우
         AddLink(d, h) ;          // d에서 h로의 연결 생성
         if (isVisited(h) == false) {
            Find_Hyperlink(h) ;
} } } return ; }
```

(그림 3.3) 웹 문서의 순환 탐색 알고리즘



(그림 3.4) 웹 문서의 구조를 표현한 그래프

(그림 3.4)는 (그림 3.1) main.html의 태그를 탐색한 결과 얻게 된 7개의 연결정보와 연결된 (그림 3.2)의 문서인 link1.html을 다시 탐색하여 얻은 3개의 연결정보를 조직화 한 결과이다. 그러나 (그림 3.4)의 정점이 표현하는 각 문서들이 실제로 링크되어 있는지 SPV만으로는 알 수 없다. 또한 문서 구조의 균형의 조정이 필요할 경우, 각 정점의 문서들이 사용 가능한 문서인지 여부를 판단하여 단절된 링크와 사용 가능한 링크를 구분하고, 각 정점들을 수정하여 3.1에서 제안한 구조화된 웹

문서 그래프를 보완한다.

3.2 단절된 링크의 탐색-SBL

웹 문서는 하이퍼링크를 통해 또 다른 웹 문서를 선형적이나 순환적으로 연결하므로 모든 하이퍼링크가 유효한 링크인지의 여부를 판단하고, 필요하다면 기존 그래프의 정점들을 수정하여야 한다. 그러나 유효성 검사 방법이 상이하기 때문에 해당 정점이 표현하는 웹 문서가 해당 웹 서버가 보유하고 있는 문서인지, 웹 서버 외부에 존재하는 문서인지의 여부 역시 구분해야 한다. 이 구분은 3.1의 구분과 마찬가지로 파일과 도메인으로 구분된다. 본 장에서는 내부 하이퍼링크와 외부 하이퍼링크의 유효성을 검사하고, 검사 방법을 기초로 3.1에서 제안한 구조화된 웹 문서 그래프를 보완한다.

3.2.1 내부 하이퍼링크의 유효성 검사

내부 하이퍼링크는 링크를 시도한 문서와 링크로 연결된 문서가 같은 웹 서버에 존재하는 경우로, 링크를 시도한 문서를 보유하고 있다면 링크로 연결된 문서 또한 보유하고 있다. 이는 HTML 파일 형태로 존재하고, 해당 파일에 접근을 요청하여 그 결과가 성공적이라면 해당 하이퍼링크는 유효하다고 판단할 수 있다. 즉, (그림 3.1)의 main.html의 링크를 탐색한 결과 얻은 7개의 링크 <표 3.1> 중, link1.html은 내부 하이퍼링크로 해당 파일이 존재한다(그림 3.2). 따라서 (그림 3.1)의 main.html에 기술된 link1.html은 유효하다고 판단할 수 있다.

(그림 3.3)의 웹 문서 순환 탐색 알고리즘을 수행하면서 해당 문서의 각 링크가 내부 하이퍼링크라 판단되면 (그림 3.5)의 알고리즘을 적용하여 해당 문서의 유효성을 검사한다. (그림 3.5)는 웹 문서가 파일일 경우에 유효성을 검사하는 알고리즘이다.

```
// 입력 : 웹 사이트의 특정 웹 문서 파일 이름 d
// 출력 : 유효한 경우(true), 유효하지 않은 경우(false)
bool Inspect_Hyperlink_File(string d) {
  // 파일의 읽기 권한을 요청
  FILE *pFileName = fopen(d, "r") ;
  // 권한 획득 실패 시 유효하지 않음
  if (pFileName == NULL) return false ;
  // 그렇지 않은 경우에는 유효함
  else return true ; }
```

(그림 3.5) 내부 문서의 유효성 검사 알고리즘

3.2.2 외부 하이퍼링크의 유효성 검사

외부 하이퍼링크는 링크를 시도한 문서와 링크로 연결된 문서가 물리적으로 다른 웹 서버에 존재하는 경우이며, 링크로 연결된 상대 문서를 보유하고 있지 않다. 따라서 다른 방법으로 하이퍼링크의 유효성을 검사해야 한다. 이러한 경우 링크는 (그림 3.1)의 main.html에서 추출된 7개의 링크 <표 3.1> 중, "http://dongguk.ac.kr"과 같은 특정 웹 서버의 도메인 주소가 된다. 이러한 경우 해당 웹 서버에 접속 가능한지 여부가 곧, 해당 링크의 유효성이다. 이 경우 특정 웹 서버로 "ping" 명령어를 사용하여 해당 웹 서버에 접속이 가능한지 여부를 판단하게 된다. "ping" 명령어는 ICMP(Internet Control Message Protocol)를 사용하는데 이 ICMP를 사용하여 웹 서버에 대한 공격이 시도되고 있어, 대부분의 웹 사이트들이 이를 차단하고 있다. 따라서 "ping" 명령어를 사용하는 데에는 어려움이 따른다.

"gethostbyname()" 메소드는 특정 웹 서버의 도메인 이름을 인자로 받아 DNS(Domain Name Server)에서 해당하는 IP주소와 Alias 등의 정보를 가져온다. 따라서 이 메소드를 이용하여 반환된 값이 없다면, 해당 웹 서버는 유효하지 않다고 판단할 수 있다.

내부 하이퍼링크의 유효성 검사와 마찬가지로 (그림 3.3)의 웹 문서 순환 탐색 알고리즘을 수행하면서 해당 문서 내부의 각 링크가 외부 하이퍼

링크라 판단되면 (그림 3.6)의 URL 유효성 검사 알고리즘을 사용하여 해당 웹 서버의 유효성을 검사한다. (그림 3.6)은 임의의 웹 문서가 URL로 주어졌을 때, 해당 문서가 유효한지 판단하는 알고리즘이다.

```
// 입력 : 웹 사이트의 특정 호스트 이름 d
// 출력 : 유효한 경우(true), 유효하지 않은 경우(false)
bool Inspect_Hyperlink_URL(string d) {
   // d 호스트의 정보를 받아 옴
   HOSTENT *ptr = gethostbyname(d) ;
   // 정보가 없다면 유효하지 않음
   if (ptr == NULL) return false ;
   else return true ;      // 그렇지 않으면 유효함
}
```

(그림 3.6) 외부 문서의 유효성 검사 알고리즘

3.2.3 구조화된 웹 문서 그래프의 갱신

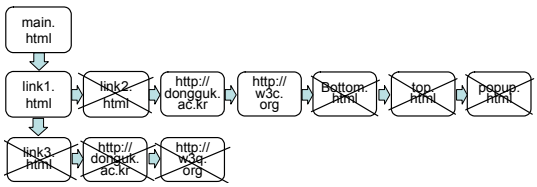
위의 하이퍼링크 유효성 검사방법을 (그림 3.3)의 웹 문서 순환 탐색 알고리즘에 적용하여 각 정점의 문서들이 사용 가능한 문서인지 여부를 판단하여 단절된 링크와 사용 가능한 링크를 구분하고, 각 정점들을 수정하여 3.1에서 제안한 구조화된 웹 문서 그래프에 유효성을 추가 한다(그림 3.8). (그림 3.7)은 임의의 웹 문서를 최초의 정점으로 출발하고 하이퍼링크만을 분석하여 연결된 모든 문서 집합을 탐색하고 해당 링크가 유효한지 검사하는

```
// 입력 : 웹 사이트의 임의의 웹 문서 d
// 출력 : AddLink()를 통해 d에서 연결된 하이퍼링크 h를 추
   출한 방향 그래프
void Find_Hyperlink(document d) {
   // 방문한 문서 d에 대해 방문 추가
   AddVisitedDoc(d) ;
   // 문서 d에 있는 각 하이퍼링크 h
   for each hyperlink h in document d {
      if(isExternal(h) == true) { // 외부 문서(URL)인가?
         // h 링크의 유효성을 저장하고 d->h 링크 생성
         AddLink(d, h, Inspect_Hyperlink_URL(h)) ;
      }
      else {   // 내부 문서인 경우
         // h 링크의 유효성을 저장하고 d->h 링크 생성
         AddLink(d, h, Inspect_Hyperlink_File(h)) ;
         if(isVisited(h) == false) {
            Find_Hyperlink(h) ;
} } } return; }
```

(그림 3.7) 개선된 웹 문서 순환 탐색 알고리즘

웹 문서 순환탐색 알고리즘이다. 이 알고리즘에서 사용된 AddVisitedDoc(), isExternal(), AddLink(), isVisited() 함수는 4.1.2와 동일하다.

(그림 3.8)은 (그림 3.7)의 개선된 웹 문서 순환탐색 알고리즘을 사용한 결과로 문서의 연결을 위한 각 정점을 추출한 (그림 3.4)에 비해 해당 링크의 유효성 검사 결과가 추가되었다. 이를 기초로 (그림 3.4)의 그래프를 (그림 3.8)과 같이 수정한다. 'X' 표기된 정점은 유효하지 않은 웹 사이트 또는 문서이다.



(그림 3.8) 웹 문서의 유효성을 추가한 그래프

4. 실험 및 모의 실험

4.1 실험

본 논문에서 제안한 웹 문서 작성기의 유지·보수성 실험은 http://sera.dongguk.ac.kr 의 모든 웹 문서를 대상으로 진행하였으며, 실험의 신뢰성을 보장하기 위해 임의의 하이퍼링크를 추가하였다.

4.1.1 웹 문서에서 하이퍼링크 추출 결과

(그림 4.1)은 웹 문서에 포함되어 있는 하이퍼링크를 통해 다른 웹 문서로 탐색하면서 하나의 웹 문서에 연결되어 있는 모든 웹 페이지를 추출한 것이다. 추출한 정점(문서)은 총 53개이며, 그 중 내부 링크는 34개, 외부 링크는 19개이다.

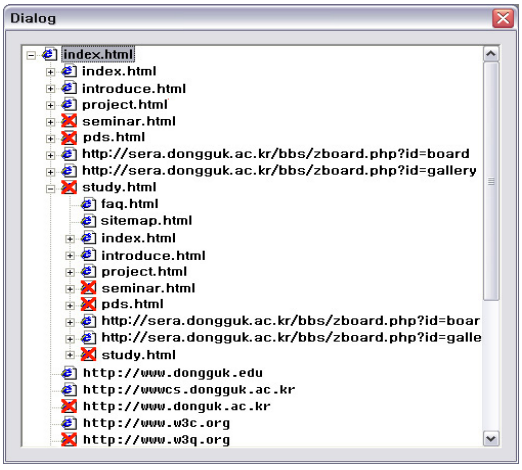
4.1.2 하이퍼링크의 유효성 검사 결과

(그림 4.1)의 추출된 하이퍼링크 중에는 실제로

유효하지 않은 하이퍼링크가 존재한다. <표 4.1>은 실제로 유효하지 않은 하이퍼링크의 목록이며, (그림 4.1)의 'X'표시는 하이퍼링크의 유효성 검사 알고리즘이 적용된 후 유효하지 않은 하이퍼링크가 표시된 결과이다. <표 4.1>의 유효하지 않은 하이퍼링크는 4개인데 (그림 4.1)에서 표시된 단절된 링크는 모두 8개이다. 이는 하이퍼링크로 연결된 페이지 내에 링크가 하나 이상 단절된 경우, 링크를 시도한 페이지에도 단절 표시를 했기 때문이다. 실제로 웹 브라우저를 통해 단절된 링크라고 판단된 URL에 대하여 접근을 시도한 결과 모두 웹 서버로의 접근이 제한되어 있었음을 확인할 수 있었다.

〈표 4.1〉 유효하지 않은 하이퍼링크의 목록

내부 하이퍼링크	외부 하이퍼링크
seminar.html	http://www.donguk.ac.kr
pds.html	http://www.w3q.org



(그림 4.1) 하이퍼링크의 추출과 단절된 링크의 탐색 결과

4.2 모의 실험(Simulation)

4.1의 실험에 사용된 http://sera.dongguk.ac.kr은 하이퍼링크 규모에 한계가 있어 더 높은 신뢰성을 증명하기 위해 보다 복잡한 웹 사이트를 모형화 하여 모의실험을 실시하였다. 그 중, 내·외부 하이퍼링크는 각각 500개씩, 문서의 깊이는 10단계로 설정하였으며, 단절된 링크는 140개로 내·외부 하이퍼링크에 각각 70개씩 설정하였다. 실험은 문서를 탐색하여 구조화 하는 실험과, 단절된 링크를 탐색하는 실험으로 구성하였다.

<표 4.2>의 실험결과를 살펴보면 실제 존재하는 총 1000개의 문서를 각 단계별로 100개씩 탐색한 결과 각 하이퍼링크에 해당하는 문서를 모두 찾을 수 있었다. 하지만, 단절된 링크를 탐색하는 과정에서 6, 7, 8, 10단계에서 각각 1, 1, 2, 2개의 잘못된 탐색 결과를 나타내는데 내부링크는 이상 없이 수행되는 데 비해, 외부링크는 일부 불일치 결과를 나타내고 있다. 이는 URL의 유효성 검사 방법이 해당 호스트의 DNS정보에 근거한 것이어서, DNS는 정보를 가지고 있으나 해당 웹 호스트

〈표 4.2〉 실험 결과

단계	문서/단절	실제 개수	외부 링크	내부 링크	일치	불일치
1	문서	100	50	50	100	0
	단절	14	7	7	14	0
2	문서	100	50	50	100	0
	단절	14	7	7	14	0
3	문서	100	50	50	100	0
	단절	14	7	7	14	0
4	문서	100	50	50	100	0
	단절	14	7	7	14	0
5	문서	100	50	50	100	0
	단절	14	7	7	14	0
6	문서	100	50	50	100	0
	단절	14	6	7	13	1
7	문서	100	50	50	100	0
	단절	14	6	7	13	1
8	문서	100	50	50	100	0
	단절	14	5	7	12	2
9	문서	100	50	50	100	0
	단절	14	7	7	14	0
10	문서	100	50	50	100	0
	단절	14	5	7	13	2

의 사정으로 인해 접근이 불가능한 경우였다. 이는 DNS에 근거한 문제점으로 새로운 보완책에 대한 연구가 요구된다.

5. 결론 및 향후과제

본 논문에서는 웹 문서 작성기의 기본 기능 외에 웹 사이트를 구조화하고 웹 문서의 하이퍼링크를 추적하여 단절된 링크를 탐색하는 방법을 제안하였다. 웹 문서의 태그 분석을 통해 하이퍼링크를 추출하여 웹을 구조화 하는 방법을 통해 연결된 모든 웹 문서들의 계층적 구조도와 함께 단절된 링크를 탐색하여 시각적으로 보여줌으로써 유지·보수 시 단절된 링크의 수정 작업을 지원하고 있다. 특히, 웹 문서 내에 포함된 하이퍼링크들 중, 링크의 단절을 해결할 경우, 웹 사이트 관리자가 일일이 하이퍼링크를 추적해 가면서 확인하여 작업하지 않고 제공된 자료를 이용하여 용이하게 해결할 수 있다. 이것은 다양한 웹 구조 개선 및 웹 마이닝을 위한 자료로 활용될 뿐 아니라 사용자들이 단절된 링크로 인해 겪는 어려움을 없애고 웹 문서의 효율적인 개발과 유지·보수를 가능하게 한다.

HTML 이외의 웹 문서를 이루는 플래시, 애플릿, 동영상 등은 링크를 형성하는 방법이 다양하고, 정확한 하이퍼링크 경로를 탐색하는데 그 구조적 특성상 어려움이 있어 이를 연구하여 보완할 필요가 있다. 또한, 하이퍼링크로 연결된 페이지들의 계층 구조를 분석하여 연결 구조가 복잡한 부분과 단순한 부분을 구분하여 관리자로 하여금 논리적인 개선을 요구하거나, 문서간의 링크 뿐 아니라 그림이나 사운드와 같은 외부 객체들의 유효성을 판단하는 등 웹 사이트를 효과적으로 유지·보수할 수 있는 방안에 대하여 연구가 이루어져야 한다.

덧붙여, 본 논문에서 제시한 방법으로 추출된 웹 문서간의 연결정보들은 하나의 웹 사이트에 한정적이다. 하지만 이러한 연결정보들을 하나의 데이터베이스로 통합하는 방안을 연구하여 웹 사이트 관리자들이 자신의 웹 사이트의 어느 부분이 다른 웹 사이트에 얼마나 많은 링크를 가지고 있는지 손쉽게 파악할 수 있게 하여 유지·보수의 전략적인 부분에도 도움을 줄 수 있는 방안을 연구하고 있다.

참 고 문 헌

[1] Jop F. Sibeyn, J. Abello and U. Meyer, “Heuristics for semi-external depth first search on directed graphs”, Association for Computing Machinery Symposium on Parallel Algorithms and Architectures, Vol. 52, pp. 282-292, 2002.

[2] Y. Kosala and H. Blockeel, “Web Mining Research, A Survey”, Newsletter of the Special Interest Group on Knowledge Discovery & Data Mining, Vol. 2, No. 1, pp. 1-15, 2000.

[3] Chakrabarti, “Mining the Web”, S. Morgan Kaufmann Pub. 2002.

[4] Thuraisingham and Bhavani M., “Web Data Mining and Business Intelligence Analysis”, CRC Press Pub. 2003.

[5] Ji-Yeon Lee, Seong-Whan Lee, “A Document-to-HTML Conversion Algorithm for Multi-Column Document Images Containing Tables”, 한국정보과학회 봄 학술발표 논문집. Vol. 26, pp. 600-602, 1999.

[6] Y. Wang and J. Hu, “Detecting Tables in HTML Documents”, Proc. 5th IAPR Int’l Workshop on Document Analysis System (DAS 2002), pp. 249-260, Princeton, USA, Aug. 2002.

[7] “DHTMLEdit SDK Technical Document”, Microsoft MSDN. 2005.
[8] “Visual C+: MFC Library Reference. 1, 2, 3”, Microsoft Corp. 1998.
[9] 최호성, “Windows programming”, Freelec. 2006.

조 영 석

1978년 서강대학교 철학과 (문학사)
1988년 Louisiana State University, MLIS(정보학)
1994년 Louisiana State University, Ph.D.(컴퓨터학)
1995년~현재 동국대학교 컴퓨터·멀티미디어학과 교수

권 용 호

2002년~현재 동국대학교 컴퓨터·멀티미디어학과 학생

도 재 수

1984년 경북대학교 전자공학과 (공학사)
1994년 일본 홋카이도 대학교
1994년 공학석사
1997년 일본 홋카이도 대학교
1997년 공학박사
1999년~현재 동국대학교 컴퓨터·멀티미디어학과 부교수