웹 페이지 관리를 위한 링크 추출과 검증

엄정섭[0] 유대승 심민석 이명재
울산대학교 컴퓨터·정보통신공학부
{quaza, ooseboss, wolf, ymj}@cic.ulsan.ac.kr

Link Extraction and Validation for Web-page Maintenance

Jeongseob Eom[0] Daesung Yoo Minsuck Sim Myeongjae Yi
School of Computer Engineering & Information Technology, University of Ulsan

요 약

웹의 발전에 따라 거대해진 웹사이트들은 서로 복잡하게 얽혀진 링크들로 인해 웹 개발과 유지보수에 큰 어려움이 따른다. 효율적인 웹 개발과 유지보수를 위해서는 웹에서 가장 중요한 정보의 단위인 링크정보들을 추출할 수 있는 방법이 요구된다. 본 논문에서는 웹 브라우저 요청에 의해 반환된 HTTP 헤더분석과 HTML 문서의 태그분석을 통해 링크들을 추출하여 "끊어진 링크"를 찾고, 추출된 "링크요소"들과 서버에 저장된 파일들을 비교하여 "사용되지 않는 파일"들을 찾아주는 "링크 분석기" 시스템을 개발함으로써 웹 개발과 유지보수에 있어서 가장 기본적이면서도 중요한 링크관리에 대한 방법을 제시한다.

1. 서 론

1992년 HTML[1]의 첫 버전이 발표된 이후 공공기관, 연구기관 및 기업에서 사용되는 문서들이 웹으로 많이 옮겨졌으며, 웹을 이용한 상업적인 컨텐츠들이 대량으로 나타나고 있다. 웹을 통해 모든 것을 할 수 있을 만큼 다양화된 방법으로 웹은 나타나고 있으며, 양 또한 점점 더 방대해지고 있다.

웹사이트 개발자들은 웹 문서들을 만들고 수정하는 과정에서 방대해진 웹 문서들을 관리하는 것에 많은 어려움을 겪고 있다. 웹사이트를 개발하거나 수정할 때 웹 문서들은 서로 복잡하게 얽혀져 있기 때문에 웹 문서들에 포함된 링크들이 제대로 연결이 되어있는지 일일이 검증하고 수정하는 것은 매우 힘들다. 이에 본 논문에서는 웹사이트에 존재하는 링크들을 관리하기 위하여 각 문서에 포함된 링크들을 추출해서 검증하는 것을 목표로 한다.

본 논문의 구성은 2장에서 기존의 연구에서 제시된 웹의 요소들을 관리하는 방법들과 이들에 대한 문제점들을 살펴보고 본 논문에서 검증하고자 하는 "끊어진 링크"와 "사용되지 않는 파일"에 대하여 설명한다. 3장에서는 본 논문에서 구현하는 "링크 분석기"에 대한 시스템에 대하여 설명하고, 4장과 5장에 이에 대한 세부적인 방법에 대하여 설명한다. 마지막으로 6장에서는 결론과 향후 연구과제에 대해서 기술한다.

2. 관련연구

2.1 XML 변환

기존의 HTML 문서들은 매우 방대한데 비해 구조화되지 못한 특징으로 관리에 어려움이 있다. W3C에서 기존의 HTML의 구조화되지 못한 단점을 보안하기 위해 XML을 제시하였다. 그 후 많은 연구에서 기존의 HTML 문서를 버리고 XML로 변환하는 방법을 제시하였다.[2]

HTML에 비해 XML은 구조화된 양식을 가지고 있기 때문에 HTML 문서를 XML로 변환하게되면 관리의 용이성이 증대된다. 하지만 기존의 HTML문서들은 너무나 방대하고, 이 문서들을 일일이 XML로 바꾸기 위해서는 많은 노력과 비용이 소요된다. HTML을 XML로 자동적으로 변환하는 도구들은 HTML의 구조를 먼저 파악해야 하기 때문에 수작업으로 변환하는 방법만큼이나 많은 노력이 필요하다.[3]

XML로 변환하는 작업은 기존의 HTML의 링크들을 XML의 구조로 매핑하는 작업이기 때문에 이러한 작업을 위해서는 HTML의 링크들을 추출해서 검증하는 작업이 먼저 이루어 져야한다.

2.2 웹 테스팅

웹의 발전과 함께 웹 기반 시스템들을 평가하기 위한 방법들이 최근 연구과제로 많이 대두되고 있다. 웹 개발의 특성상 웹 기반 시스템은 데이터베이스, CGI, C++, JAVA, ASP, JSP, DCOM, EJB등 단일 개발에 대해 여러 가지의 언어와 기술이 통합적으로 사용된다는 점에서 웹에 대한 테스팅은 일반적인 시스템들과는 다른 방법으로 다루어진다.

웹 테스팅에서는 각 웹 문서들을 서로 연결시켜주는 링크들이 일반적인 어플리케이션의 데이터 흐름에 해당한다.[4][5] 따라서 웹 기반 시스템을 평가하기 위해서는 링크들의 추출이 먼저 이루어져야한다.

2.3 "끊어진 링크"와 "사용되지 않는 파일"

웹 문서들을 서로 연결해 주는 링크들은 태그에 의해 표현된다[6]. 태그를 이용해 연결할 수 있는 링크들은 일반적인 웹 문서뿐만 아니라 이미지, 사운드, 애니메이션 등 멀티미디어 파일과 자바애플릿, E-Mail, FTP와 같은 특수한 경우도 존재한다.

웹사이트 내부에는 잘못된 링크들이 포함될 수 있다. 웹 페이지 추가/삭제와 같은 수정 작업 시에 웹 페이지에는 올바르지 않은 링크들이 발생한다.

그림 1의 (a)에서와 같이 웹 문서인 Page3을 삭제할 때, Page2의 Link2는 대상이 없는 링크가 되고, (b)에서와 같이 Page2를 가리키는 Page3을 추가할 때, Page3은 새로 생성된 존재하는 문서이지만 Page3을 가리키는 링크가 없기 때문에 사용자 측에서는 실제로 도달할 수 없는 문서가 된다. 기존의 연구에서 (a)와 같이 대상이 없는 잘못된 링크를 "Dangling

본 연구는 정보통신부의 "정보통신 우수시범학교 지원사업"의 지원에 의해 이루어졌습니다.

Link"라하고, (b)와 같이 존재하지만 볼 수 없는 페이지들을 "Unreachable Webpage"라고 정의해 놓았다.[7] 본 논문에서는 전자를 "끊어진 링크", 후자를 서버에는 존재 하지만 사용되지 않는 요소이므로 "사용되지 않는 파일"이라고 정의한다.

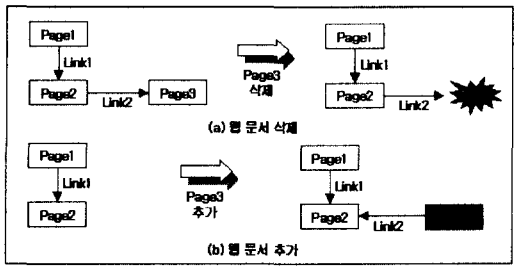

그림 1 웹 문서의 수정

3. 링크분석 시스템

본 논문에서 구현하는 "링크 분석기"는 그림 2에서처럼 웹사이트 개발자나 관리자와 같은 사용자들이 사이트의 최초 페이지, 즉 해당 사이트의 제일 먼저 보게되는 페이지를 입력함으로써 해당 사이트의 모든 링크정보를 얻고, 이 과정에서 "끊어진 링크"와 "사용되지 않는 파일"들을 식별해 낸다.

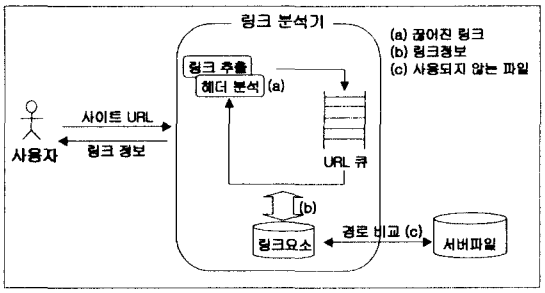

그림 2 시스템 구조도

4. 링크 추출과 검증

4.1 태그 분석을 통한 링크 추출

링크를 추출하는 방법으로는 서버에 저장된 문서들의 분석을 통한 방법과 웹 브라우저의 요청으로 서버로부터 반환된 HTML 문서들의 분석을 통한 방법이 있다. 전자의 경우 동적인 웹 페이지에 존재하는 링크 추출이 어렵다. 따라서 본 연구의 "링크 분석기"는 웹 브라우저의 요청에 의해 반환된 HTML 문서들을 분석하는 과정을 통하여 링크들을 추출한다.

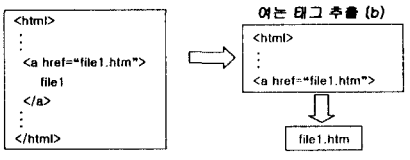

그림 3 태그로부터 링크 추출

그림 3에서와 같이 웹 서버에서 반환된 HTML 문서부터 <html>, <a href="file1.htm">등 시작 태그를 우선 추출해서 태그가 <a>와 같이 링크를 포함하는 태그인지 판별한 뒤, 태그의 속성으로부터 "file1.htm"인 링크 값을 추출해 낸다.

4.2 헤더 분석을 통한 링크 검증

추출되어진 링크들은 HTTP 헤더 분석을 통해 올바른 링크인지를 판별하여 "끊어진 링크"를 찾아낸다.

HTTP 헤더에는 그림 4와 같이 웹 문서에 대한 다양한 정보들을 담고 있다.

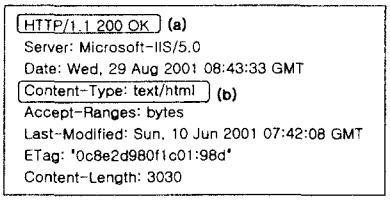

그림 4 HTTP 헤더

그림 4의 (a)는 HTTP의 상태를 나타낸다. 그림에서 보여지는 "200"이라는 숫자는 올바른 링크임을 나타내는 상태코드를 뜻한다[8]. "403 Access Forbidden"과 "404 Object Not Found"와 같이 에러코드를 반환하거나 서버를 찾을 수 없거나 DNS 오류가 날 경우 "링크 분석기"는 "끊어진 링크"로 처리한다.

(b)는 MIME 타입을 나타내는데 "링크 분석기"에서는 이것을 웹 문서와 일반적인 파일들을 구분하는데 사용한다.

4.3 링크 추출 과정

링크 추출 과정은 그림 5와 같은 절차를 거친다.

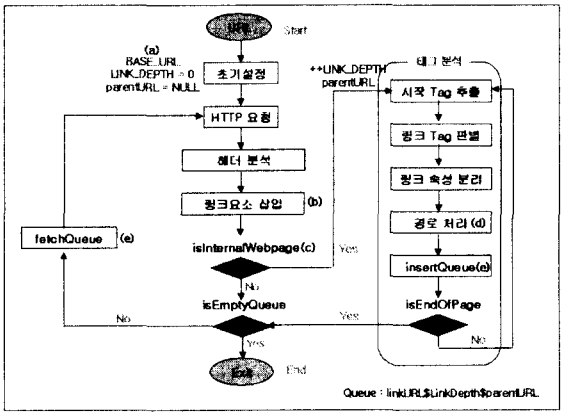

그림 5 링크 추출 과정

(a) 초기설정

초기 웹 문서의 경로(URL)가 입력되면 그 경로를 통해서 기본 URL(BASE_URL)과 링크의 깊이(LINK_ DEPTH), 상위경로(parentURL)가 정해진다. BASE_URL은 뒤에 설명되는 "내부 웹 페이지"구분과 링크의 경로 처리에 이용된다.

초기 linkDepth는 "0"으로 처리하고, parentURL은 최상위 경로이기 때문에 "NULL"로 처리한다.

(b) 링크요소 삽입

링크 추출 과정에서 추출된 모든 링크들은 표 1과 같은 필드

를 가진 데이터 형식으로 저장되고, "링크 분석기"에서는 이러한 링크의 정보를 가진 하나의 레코드를 "링크요소"로 정의한다.

표 1 링크요소

필드명	설명
linkURL	추출한 링크의 전체경로를 나타내는 문자열, 유일한 값
linkStatus	HTTP 헤더 분석을 통해 반환되는 값을 저장하는 문자열
linkContentType	링크의 MIME 타입을 저장하는 문자열
linkDepth	초기에 입력된 링크경로에서부터 몇 번째 추출되었는지 나타내는 정수값
parentURL	링크가 어느 문서에서부터 추출되어 졌는지를 나타내는 문자열

linkStatus와 linkContentType은 헤더 분석을 통해 얻어지고 나머지 세 개의 필드들은 초기설정과 태그 분석 과정에서 정해진다.

(c) 내부 웹 페이지

"isInternalWebpage" 과정은 처리중인 링크요소가 하위에 다른 링크요소들이 포함되어져 있는지를 구분하는 절차이다.

링크요소들 중에는 단순한 이미지파일 같이 다른 링크요소들을 포함하지 않는 경우와 "linkContextType"이 "text/html"일 때와 같이 다른 링크요소들을 포함하는 경우로 나뉜다. 다른 링크요소를 포함하는 경우라도 해당 링크가 외부 사이트를 연결하는 링크일 경우에는 계속적인 분석이 필요하지 않으므로 1차 검증후 하위적인 분석 과정은 제외한다. 따라서 "isInternalWebpae"는 처리하는 링크가 내부 링크이고, 웹 문서인 경우일 때만을 가려내어 태그 분석 과정으로 넘어간다.

(d) 경로처리

4.1장에서 설명한 방법을 통해 추출된 링크들은 상대경로와 같이 완전하지 않은 경로를 포함하는 경우가 있기 때문에 이것을 완전한 경로로 바꾸어주는 과정이다.

(e) 큐 삽입과 가져오기

태그 분석 과정을 거쳐서 얻어진 링크들은 다시 검증을 위해 임의의 저장소인 큐에 삽입한다.

큐의 데이터 형식은 링크의 경로(linkURL)외에 헤더분석을 통해 얻어지지 않는 "LinkDepth"와 "parentURL" 필드 값이 함께 저장한다. 이렇게 저장된 큐는 데이터를 가져오는 과정에서 다시 분리되어 링크요소의 각 필드에 저장된다.

5. 사용되지 않는 파일 추출

"사용되지 않는 파일"들은 그림 6과 같은 방법으로 추출한다.

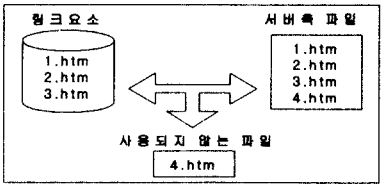

그림 6 사용되지 않는 파일

"사용되지 않는 파일"이란 서버 측 자원으로 존재하면서도 "링크요소"로 나타나지 않는 파일이므로 링크 추출 과정을 통해 얻어진 "링크요소"들과 "서버에 저장된 파일"들을 비교하여 "서버에 저장된 파일"들 중 "링크요소"에는 없는 파일들을 하나씩 제거해 나감으로써 "사용되지 않는 파일"들을 찾는다.

6. 결론 및 향후과제

웹 개발과 유지 보수에 있어서 웹 페이지에 포함된 링크들은 웹 시스템의 중요한 정보의 단위에 해당하며, 웹사이트에 포함될 수 있는 "끊어진 링크"와 "사용되지 않는 파일"들은 웹사이트 링크 관리에 있어서 가장 큰 문제가 될 수 있는 요인이다.

본 연구에서는 HTTP 헤더분석을 통해 링크들을 검증하여 "끊어진 링크"를 찾고, 링크추출을 통해 얻어진 "링크요소"들과 "서버에 저장된 파일"들을 비교하여 "사용되지 않는 파일"들을 찾아주는 "링크 분석기"를 구현함으로서 효율적인 링크정보에 대한 관리 방법을 제시하였다.

본 연구에서 얻어진 링크들에 대한 정보들을 바탕으로 향후 연구과제로 사이트 구조화와 같은 웹 가시화에 대한 연구가 진행될 수 있다. 또한 링크들에 대한 정보들은 웹 테스팅에 대한 중요한 기반 자료가 될 것이다.

참고문헌

[1] Hypertext Markup Language Home Page, http://www.w3.org/MarkUp

[2] L.Bishay, J.W.Rahayu, D.Taniar, "Transformation from HTML to XML: Methodology and Tool", submitted for publication, 2000.

[3] D.Taniar, Y.Jiang, J.W.Rahayu, L.Bishay, "Structured Web Pages Management for Efficient Data Retrieval", Web Information Systems Engineering, vol.2, Page(s): 97 -104, 2000

[4] Ji-Tzay Yang, Jiun-Long Huang, and Feng-Jian Wang, "A Tool Set to Support Web Application Testing", International Computer Symposium, December 1998.

[5] Ji-Tzay Yang, Jiun-Long Huang, Feng-Jian Wang, "An Object-Oriented Architecture Supporting Web Application Testing", Computer Software and Applications Conference, On page(s): 122 - 127, 1999.

[6] HTML 4.0 Specification, http://www.w3.org/TR/1998/REC-html40-19980424

[7] Chia-Lin Hsu and Feng-Jian Wang, "A Web Data_base Application Model for Software Maintenance", National Chiao-Tung Univertisy, Master Thesie, 1998.

[8] Hypertext Transfer Protocol -- HTTP/1.1, http://www.w3.org/Protocols/rfc2616/rfc2616.txt