제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

웹 사이트의 효율적인 구조 관리와 평가를 위한 시스템의 설계 및 구현

김종환*,유대승, 박재희, 이명재
울산학교 컴퓨터 · 정보통신공학부
e-mail:bearknight@gmail.com

A Design and Implementation of System for Efficient Structure Management and Evaluation of Website

Kim Jong-Hwan*, Dae-Sung Yoo, Jae-Hee Park, Myeong-Jae Yi

School of Computer Engineering & Information Technology,
University of Ulsan

요 약

인터넷과 웹의 급속한 성장과 함께 기존의 시스템들이 웹을 기반으로 통합되면서 비즈니스 환경을 급속하게 변화시키고 있다. 그러나 웹은 다양한 기술의 접목으로 개발된다는 점과 본래의 복잡성으로 인해 개발과 관리에 있어 어려움이 더욱 증대되고 급변하는 비즈니스 환경과 사용자들의 요구사항에 순응하기 위해서는 지속적인 진화가 요구된다. 이와 같은 웹의 본질적인 복잡함과 짧은 생명주기는 웹 사이트의 성공적인 개발과 유지보수를 위해서 많은 비용과 노력의 소요를 유발한다. 본 논문에서는 웹 사이트의 개발과 유지보수에 대한 비용을 절감하고 효율적인 구조 관리와 평가를 지원하기 위해 개발된 시스템을 제안한다.

1. 서론

인터넷과 웹의 빠른 발전과 더불어 비즈니스 프로세스, 지식 경영, 마케팅, 전자 상거래 등 현대의 다양한 IT 시스템들이 웹으로 통합되면서 비즈니스 환경을 급격하게 변화시키고 있다. 그러나 웹 통합 개발 과정에 있어 필연적으로 파생되어진 복잡함은 개발 및 유지보수를 더 어렵게 하며, 이에 따른 비용 또한 증가하고 있다[1][2]. 인터넷과 관련된 기업의 폭발적인 증가와 함께 성공적인 웹 사이트 운영을 위해서는 빠르게 변화하는 환경과 사용자들의 요구사항에 순응하며 서비스를 제공하는 시스템을 지속적으로 진화시켜야 한다. 웹 사이트의 본질적인 복잡함과 짧은 생명주기로 인하여 웹 사이트의 성공적인 개발과 유지보수를 위한 많은 비용과 노력이 소요되는 상황에서 웹 사이트의 효율적인 관리와 평가 방법 및 시스템적인 지원은 웹 사이트의 개발과 유지보수에 있어 핵심적인 요소라 할 수 있다.

본 논문에서는 효율적인 구조 관리와 평가를 위한 시스템을 제안한다. 본 논문에서 제안하는 시스템은 기존의 웹 사이트로부터 다양한 정보들을 추출하여 저장한 후 웹 사이트의 관리와 평가에 활용 할 수 있도록 가공해서 보여준다. 또한 이 정보들은 웹 사이트의 개발과 유지보수 시에 유용하게 사용되어질 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 연구에서 제시된 웹 사이트 분석 방법들과 도구에 대하여 살펴본다. 3장에서는 본 논문에서 제안하는 웹 사이트의 구조 분석과 평가 시스템을 소개한다. 4장에서는 시스템의 실행화면과 메뉴 구성을 보인다. 마지막으로 5장에서는 결론과 향후 연구과제에 대해서 기술한다.

2. 관련연구

일반적인 웹 사이트들은 시장의 요구에 따라 매우 짧은 개발 일정과 빈번한 개발자 교체, 그리고 웹 개발 기술의 빠른 발전으로 인해 구조화와 문서화가 부족한 실정이다[3]. 이러한 문제들을 해결하기 위해 기존 연구에서는 방법론, 모델, 프로세스와 도구들을 이용한 측면에서 여러 가지 방법들을 제시하고 있다. 본 논문에서는 방법론과 도구로 나누어서 기술하도록 한다.

2.1 방법론

Antoniol등은 Relation Management Methodology (RMM)를 이용하여 웹 어플리케이션 구성 요소를 엔티티 관계 모델로 제시하였다[4]. RMM을 통해 웹 어플리케이션을 개발하고 유지보수 하는데 있어 개발자와 관리자의 어플리케이션에 대한 이해를 높여 웹 어플리케이션에 부족한 구조화와 문서화의 취약점을 보완할 수 있다.

Ceri등은 Web Modeling Language(WebML)를 제시하였다[5]. WebML은 웹 어플리케이션의 상위 수준의 개념적인 설명을 제공한다. Conallen은 웹 문서들을 UML 구성요소로 모델링 하는 Web Application Extension(WAE)를 제시하였다[6]. WebML과 WAE는 기존에 이미 만들어진 시스템에 이 모델을 적용하기 위해 기존의 시스템을 분석하는 과정을 필요로 하며 자료의 흐름을 모델링 하는 측면이 부족하므로 실제적인 구현을 모델링 하는 것보다 웹 어플리케이션의 명세서에 더 적합하다.

Boldyreff등은 웹 사이트를 분석하여 중복된 웹 컨텐츠와 스타일을 추출하는 방법을 제안하였다[3]. 기존의 역 공학을 이용하여 웹 어플리케이션의 모든 요소들을 분리하여 하나의 저장소에 저장하고 이것을 다시 하위에서 세부적인 정보로 나누어서 기존의 웹 개발과 유지보수 방법보다 효율적인 정보로 새롭게 만들어 낼 수 있다.

2.2 도구

Brereton등은 HTML의 태그를 프로그래밍 언어의 블록 구조와 유사하다고 지적하고, 웹 문서 사이를 연결하는 링크들은 GOTO 문장과 유사하다고 지적하였다[7]. 그들은 웹 사이트에 존재하는 문서들을 평가할 수 있는 도구를 연구하였다. 그들이 개발한 도구는 일정 기간동안 웹 사이트를 방문하여 웹 문서 내에서 변경된 점을 보고해 준다. Ricca와 Tonella도 이와 유사한 도구를 개발하였다[8].

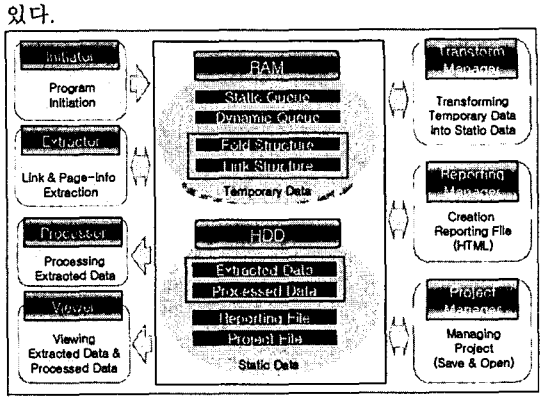
상용의 웹 구조분석을 위한 도구들 중에서 다른 것들과 비교되는 특징을 가진 것으로는 REL Software[9]의 "Web Link Validator"와 KyoSoft[10]의 "Link Checker Pro"가 있다. "Web Link Validator"는 웹 사이트의 링크 검사와 관리를 위한 도구로 끊어진 링크와 함께 구문 오류를 포함하는 링크까지 추출해 주는 장점이 있다. 그러나 정적인 문서에만 초점이 맞추어져 있기 때문에 동적인 웹 문서를 포함하는 사이트의 구조정보를 추출하지 못하는 한계가 있다. "Link Checker Pro"는 웹 사이트의 분석을 위한 도구로 웹 사이트 내의 전체 링크들의 연결 상태를 그래프로 표현하는 장점을 가지는 반면 분석 속도는 "Web Link Validator"에 비해 느리다.

3. 웹 사이트의 구조 분석과 평가 시스템

본 논문에서 제안하는 시스템은 웹 사이트 개발자나 관리자와 같은 사용자들이 사이트의 시작 페이지를 입력함으로써 해당 사이트의 구조 정보라고 할 수 있는 사이트 내의 모든 링크 정보와 사이트의 평가와 유지보수에 활용할 수 있는 여러 가지 유용한 정보들을 제공 한다.

3.1 시스템 구성

본 논문에서 제안하는 시스템은 4개의 모듈, 3개의 매니저와 저장소들로 구성된다. 다음의 [그림 1]은 시스템의 구성과 각 구성 요소들의 주요 기능을 보이고 있다.

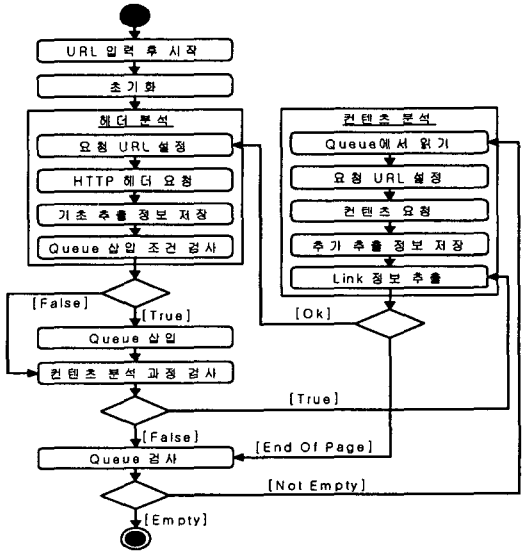


[그림 1] 시스템 구성도

Extractor 모듈에서 추출된 정보는 주 기억장치의 임시 저장소에 저장된 후 Processor 모듈에 의해 가공되고 Viewer 모듈에 의해 사용자에게 다양한 형태로 보여주게 된다. Transform 매니저는 임시 저장소에 저장된 정보와 Processor 모듈에서 가공된 정보를 XML 문서 형식으로 변환하여 보조기억장치인 정적 저장소에 저장한다. Reporting 매니저는 추출되고 가공된 정보를 HTML 형태로 변환된 파일로 저장하여 시간과 장소에 제한 없이 웹브라우저를 통해서 볼 수 있도록 한다.

3.2 웹 사이트로부터 정보 추출

다음의 [그림 2]는 웹 사이트로부터 정보 추출 과정을 활동 다이어그램으로 보이고 있다.



[그림 2] 웹 사이트로부터 정보추출 과정

웹 사이트로부터 정보의 추출은 시작 페이지에 대한 URL 입력으로부터 연결된 웹 사이트내의 모든 링크들에 대한 정보 추출이 완료될 때까지 헤더 분석과 컨텐츠 분석의 반복적 추출 작업을 통해서 이루어진다. 헤더 분석은 요청 URL에 대한 HTTP 헤더 요청을 수행하고 반환된 헤더에서 추출할 수 있는 정보들(상태코드, Content-Type, Last-Modified, Content-Length 등)을 추출한 후 Queue 삽입 조건을 검사한다.

다음의 [그림 3]은 HTTP 헤더 요청에서 반환된 헤더 정보의 예를 보여준다.

```
HTTP/1.1 200 OK (a)
Server: Microsoft-IIS/5.0
Date: Wed, 7 Sep 2004 05:43:33 GMT
Content-Type: text/html (b)
Accept-Range: bytes
Last-Modified: Wed, 4, Oct 2004 07:42:05 GMT
ETag: "0c8e2d980f1c01:98d"
Content-Length: 3030
```

[그림 3] 반환된 HTTP 헤더

[그림 3]의 (a)는 HTTP의 상태를 나타낸다. "200"은 올바른 링크임을 나타내는 상태코드를 뜻한다. "403 Access Forbidden"과 "404 Object Not Found"와 같이 에러코드를 반환하거나 서버를 찾을 수 없거나 DNS오류가 날 경우 "끊어진 링크"로 처리한다.

(b)는 MIME 타입을 나타내는데 이것을 링크를 포함할 수 있는 웹 문서와 일반적인 파일들을 구분하는데 사용한다.

Queue는 향후 컨텐츠 분석이 필요한 URL들을 저장하는 곳으로 Queue가 비었다는 것은 더 이상 분석할 URL이 없다는 것을 의미함으로 종료 조건으로도 사용된다. 실제 구현에서는 [그림 1]과 같이 "Static Queue"와 "Dynamic Queue"로 분리하여 정적문서에 대한 URL은 "Static Queue"에 동적문서에 대한 URL은 "Dynamic Queue"에 저장하여 "Static Queue"와 "Dynamic Queue"가 모두 빈 경우 종료상태가 된다.

Queue의 삽입조건은 이전에 컨텐츠 분석을 수행하지 않은 URL이어야 하고 시작페이지의 URL로부터 파생된 내부 URL이어야 한다. 또한 반환된 헤더의 상태 코드가 "2xx"와 같이 요청이 성공인 경우가 되어야 하며 컨텐트 타입이 "text"와 같이 하위 링크를 포함 가능해야 한다.

컨텐츠 분석은 요청 URL에 대한 실제 문서의 내용을 요청하고 반환된 문서에서 추가 추출 정보들(문서 제목, 문서 크기, 다운로드 시간, 링크가 포함된 태그 등)을 저장하고, 문서 내에 포함된 하위 링크 정보들을 추출하는 과정이다.

컨텐츠 분석에서 하위 링크 정보를 추출하기 위해서는 링크를 포함할 수 있는 태그와 속성이 정의되어야 하며 다음의 [표 1]은 이것을 보여준다.

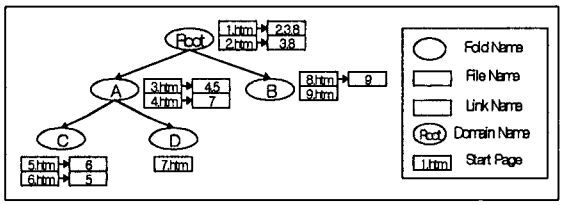
[표 1] 링크를 포함하는 HTML 태그와 속성

(속성 뒤의 "*"은 필수 속성을 의미)

HTML Tag	Attribute
<A>	href
<APPLET>	codebase, code*
<AREA>	href
<BASE>	href
<BLOCKQUOTE>,<Q>	cite
<BODY>	background
<BGSOUND>	src
<DEL> or <INS>	cite
<EMBED>	src
<FORM>	action
<FRAME>, <IFRAME>	longdesc, src
<HEAD>	profile
<IMG>	longdesc, src*, usemap
<INPUT>	src, usemap
<LAYER>, <ILAYER>	background, src
<LINK>	href
<META>	content
<OBJECT>	classid, codebase, data, usemap
<PARAM>	value*
<SCRIPT>, <JAVASCRIPT>	src
<TABLE>, <TD>, <TH>, <TR>	background

3.3 추출된 정보의 저장

다음의 [그림 4]는 추출된 정보의 저장 예를 보이기 위하여 간단한 웹 사이트의 구성을 보였다. [그림 5]는 [그림 4]에 대하여 추출된 정보의 저장 예이다.



[그림 4] 사이트의 구성 예

Fold Structure

LC : Last Child PS : Previous Sibling LL : Last Link

Index	Fold Name	Parent Index	LC Index	PS Index	LL Index
1	Root	-1	3	-1	2
2	Root/A	1	5	-1	5
3	Root/B	1	-1	2	7
4	Root/A/C	2	-1	-1	9
5	Root/A/D	2	-1	4	8

Link Structure

Index	Previous Index	Info
1	-1	1.htm
2	1	2.htm
3	-1	3.htm
4	-1	8.htm
5	3	4.htm
6	-1	5.htm
7	4	9.htm
8	-1	7.htm
9	6	6.htm

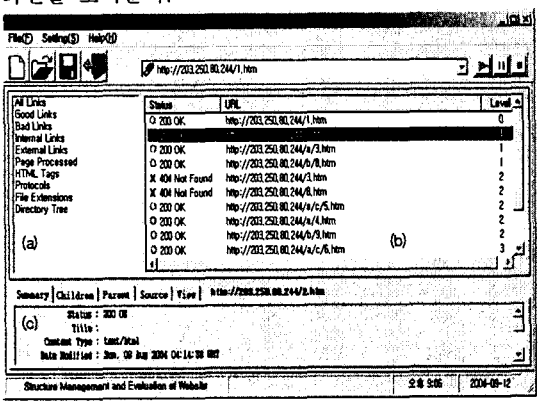
[그림 5] 추출 정보의 메모리 저장 예

웹 사이트의 구조는 일반적으로 복잡한 계층적인 구조로 형성되며 링크 정보의 추출 시 이미 추출된 링크에 대한 반복적인 처리를 피하기 위해서 새롭게 추출된 링크에 대하여 이미 저장되어 있는지에 대한 존재여부를 검색하게 된다. 이때 새롭게 추출된 링크는 일반적인 검색에 비해 검색 실패 확률이 높으므로 비교 횟수를 줄이는 것이 속도 향상의 중요한 요소이다. 따라서 본 논문에서는 폴더 단위로 링크 정보를 저장하고 폴더 내의 링크들은 연결 리스트를 구성해서 링크의 존재 여부를 폴더 내에서만 검색함으로써 비교 횟수를 줄이고 있다. 또한 폴더 구조와 링크 구조를 연결 리스트로 구성하여 축척된 정보를 쉽게 계층적 구조로 재구성 할 수 있도록 하였다.

4. 시스템 구현

본 논문에서 구현한 시스템은 Visual Basic 6.0 환경에서 컨텐츠 분석과 문서 변환을 위하여 XML 파서로 MSXML 3.0을 사용하였다.

다음의 [그림 6]은 본 논문에서 구현한 시스템의 실행화면을 보여준다.



[그림 6] 실행화면

[그림 6]의 (a)는 추출되고 가공된 정보들에 대한 10개의 메인 메뉴로 트리 구조로 구성되며 다음은 각 메뉴의 항목과 간단한 설명을 보인다.

- All Links : 추출된 모든 링크를 트리 구조로 보여준다.
- Good Links : 상태코드 4xx, 5xx를 제외한 정상적인 링크
- Bad Links : 상태코드 4xx, 5xx의 비정상적인 링크
- Internal Links : 시작 URL과 동일한 도메인을 가진 링크
- External Links : 시작 URL과 다른 도메인을 가진 링크
- Page Processed : 4가지(Slow, Small, New, Old)로 분류
- HTML Tag : 링크 정보가 추출된 태그 별로 분류
- Protocols : 프로토콜별로 해당하는 링크들을 분류
- File Extension : 파일 확장자 별로 링크(파일)를 분류
- Directory Tree : 시작 URL의 도메인을 Root로 한 폴더 구조

[그림 6]의 (b)는 (a)에서 선택한 메뉴에 해당하는 링크들을 리스트 형태로 보여주며 (c)는 (b)에서 선택한 링크에 대한 상세한 정보를 보여준다. (c)는 5개의 탭(Summary, Children, Parent, Source, View)으로 구분된다. Summary 탭에서 선택한 링크에 대하여 9가지 항목(Status, Title, Content Type, Date Modified, Size, Response Time, HTML Tag, Level, Internal)으로 분류해서 보여준다. Children 탭에서는 선택한 문서에 포함된 링크들을 보여주고 Parent 탭에서는 선택한 문서에 대한 링크를 포함하는 링크들을 보여준다. Source 탭은 선택한 문서의 HTML 소스를 보이고 View 탭에서 선택한 문서의 웹브라우저 뷰를 보인다.

5. 결론 및 향후 연구과제

본 논문에서는 효율적인 구조 관리와 평가를 위하여 개발된 시스템의 구현을 보였다. 본 논문에서 헤더 분석과 컨텐츠 분석을 통하여 웹 사이트로부터 정보를 추출하는 과정에 대하여 기술하였고 링크를 포함하는 태그와 속성을 정리하여 보였다. 그리고 효율적인 링크 정보 검색과 추출된 정보의 재구성을 위한 저장구조에 대하여 설명하였다. 마지막으로 본 논문에서 소개한 시스템의 실행화면을 보이면서 웹 사이트에서 추출 가능한 정보들의 항목들과 이들을 분류해서 메뉴화한 방법에 대하여 기술하였다.

본 논문에서 기술한 시스템은 웹 사이트의 개발자들에게는 개발과 유지보수에, 웹 사이트 관리자들에게는 관리와 평가에 유용하게 사용될 수 있다.

향후에는 웹 사이트에서 추출한 여러 가지 정보들의 활용도를 높이기 위하여 구조적이면서 쉽게 이해될 수 있는 표현 방법에 대한 추가적인 연구와 다양한 뷰를 제공하는 방법들에 대한 연구가 필요하다.

참고문헌

[1] Benoit Leger, Jean-Christophe Cimetiere, "Web Load and Performance Testing Tools", "www.trendmarkers.com", 2000

[2] Hung Q. Nguyen, "Testing Applications on the Web", Wiley Computer Publishing, 2001

[3] Boldyfeff C., Kewish R., "Reverse engineering to achieve maintainable WWW site", Reverse Engineering, 2001. Eighth Working Conf. on, 2001 [pp. 249 - 257]

[4] G. Antoniol, G. Canfora, G. Casazza, A. D. Lucia, "Web Site Reenginnering using RMM", In Procceedings of euroREF: 7th Reengineering Forum, Zurich, Switzerland, Mar, 2000.

[5] S. Ceri, P. Fraternali, A. Bongio. "Web Modeling Labguage(WebML): a modeling language for designing Web sites", In The Ninth International World Wide Web Conference, Amsterdam, Netherlands, May 2000

[6] J. Conallen, "Building Web Application with UML", object technology, Addison-Wesley Longman, Massachusetts, USA, first edition, Dec 1999

[7] P. Brereton, D. Budgen, G. Hamilton., "Hypertext: The Next Maintenance Mountain", Computer 31(12):49-55, Dec, 1998

[8] F. Ricca and P. Tonella, "Visualization of Web Site History", In Proceedings of euroREF: 7th Reengineering Forum, Zurich, Switzerland, Mar. 2000.

[9] http://www.relsofware.com

[10] http://www.kyosoft.com