웹 사이트에서 하이퍼링크의 무결성 유지를 위한 WPMS에 관한 연구

조이기*, 이성재**, 박나연**, 김원중*
*순천대학교 컴퓨터과학과
**순천제일대학

e-mail:kwj@sunchon.ac.kr

A Study on WPMS for Integrity Preservation of Hyperlink in Web Site

Lee-gi Cho*, Sung-jae Lee**, Na-yeon Park**, Won-jung Kim*
*Dept of Computer Science, Sunchon National University
**Suncheon First College

요 약

TCP/IP 프로토콜을 기반으로 한 웹은 컴퓨터 기술과 통신 기술의 발달과 더불어 양적으로나 복잡도에 있어서 놀라운 속도로 성장하고 있다. 이에 따라 웹 사이트를 이용한 홍보나 기업의 이익을 목적으로 하는 웹 사이트들이 급속히 증가하면서 웹 페이지들의 추가, 삭제, 갱신 등 끊임없이 일어나고 있다. 이렇게 끊임없는 웹 페이지의 수정으로 인하여 링크가 끊어지는 등의 문제가 발생할 수 있는데, 이런 끊어진 링크의 발견은 쉽지 않고, 때에 따라서는 기업에 큰 손실을 가져올 수 있다.

본 논문에서는 웹 페이지 문서들 사이에 존재하는 관계성(Relationship)과 제약 조건(Constraint Condition)을 확장 UML을 통해 정의하여, 웹 페이지의 DLP(Dangling Link Problem)를 해결하기 위한 WPMS(Web Page Integrity Management System) 시스템을 제안하였다.

1. 서 론

HTML을 기반으로 하는 WWW의 웹 페이지에서 하이퍼링크의 선언과 스타일의 포함은 매우 중요하다[1]. 그러나 하이퍼링크를 선언하는데 있어서 데이터베이스와 같이 개체와 개체를 연결하는데 있어 제약 조건을 부여하지 않고 단순하게 연결만을 정의하기 때문에 링크가 잘못 연결되거나, 끊어지는 경우가 자주 발생하게 되는데 일반적으로 이러한 현상을 DLP(Dangling Link Problem)라고 한다[2,3]. 최근에는 문서 안에 정보를 표현하는 스타일이 포함되어 사용자가 직접 작성한 웹 페이지더라도 DLP의 발견이 쉽지 않다[4,5].

깨진 링크의 발견 및 수정을 위해 사용되고 있는 기존의 방법들은 원하는 웹 페이지에 접근하고자 할 때 새로운 문서를 가르키도록 페이지를 포워딩하는 방법과 웹 페이지의 변동시 변동상황을 사용자에게 알림으로서 이동경로를 정확하게 파악하는 방법 등이 있다.

본 논문은 정보통신부지원 2003년도 대학기초연구지원사업(정보통신기초기술연구지원사업)의 결과임

이들 방법의 경우는 웹 페이지들 간의 링크의 깨짐을 해결하기보다는 단순히 URL의 위치를 변경시켜 문제를 해결하는 방법이라 볼 수 있다. 이처럼 단순한 URL의 위치 변경이 아니라 웹 페이지의 모든 링크 중에 링크의 연결이 깨질 경우 DLP를 추출하여 링크의 제약조건에 따라 링크를 변경함으로서 'HTTP 404 찾을 수 없습니다'라는 경고 메시지가 나타나지 않도록 사전에 예방하여 사용자에게 신뢰감을 주는 웹 페이지를 만들 필요성이 크게 대두되고 있다[6].

본 논문에서는 이러한 DLP의 발생을 최소화하기 위해 웹 페이지 설계 단계에서부터 문제 해결을 위한 접근 방법을 적용하는 방안을 제시하였다.

본 논문의 구성은 2장에서 웹 사이트에서의 참조무결성 정의 및 적용에 대해 설명하였고, 3장에서는 DLP 해결을 위해 실제 WPMS를 설계 구현함으로써 어떻게 웹 페이지의 무결성을 유지할 수 있는가에 대해 자세히 알아본다. 4장에서는 기존의 방식을 이용한 웹 페이지 개발과 WPMS을 이용한 웹 페이지 개발의 성능분석 결과를 제시하였다.

2. 웹 사이트에서의 참조무결성 정의 및 적용

DB에서의 참조 무결성은 Parent와 Child에서 삭제, 삽입과 갱신이 동시에 발생될 수 있지만, 웹 사이트에서의 참조 무결성은 [표 1]과 같이 Parent와 Child에서 삭제에 관련된 내용에 관해서만 적용을 하고, 삽입이나 갱신의 경우 웹 페이지 무결성 유지와는 거리가 멀어 본 논문에서는 고려하지 않았다[7].

[표 1] 웹 사이트에서의 참조 무결성 조건

	Delete	Insert	Update
Parent	Restrict	-	-
	Cascade	-	-
	None	-	-
Child	Restrict	-	-
	Cascade	-	-
	-	-	-

PDR(Parent Delete Restrict)이란 Parent 페이지의 링크를 삭제하면 Child 페이지가 고아 페이지가 되는 경우로, Child 페이지만으로는 의미가 없는 제약조건을 말한다. PDC(Parent Delete Cascade) 이란 Child 페이지가 독립적으로는 의미가 없어 Parent 페이지의 링크 삭제시 Child 페이지를 동시에 삭제하도록 하도록 하는 제약조건을 말한다. PDN(Parent Delete None)이란 두 개 이상의 Parent가 하나의 Child 페이지를 가르키고 그중 관계를 추후에 삭제해도 문제가 없을 경우에 적용하도록 하는 제약조건을 말한다.

CDR(Child Delete Restrict) 이란 가장 일반적인 경우로 Child 페이지를 삭제할 경우 Parent 페이지의 링크가 깨지는 DLP가 발생하게 됨에 따라 참조되고 있는 Child 페이지의 삭제를 제한하도록 하는 제약조건을 말한다. CDC(Child Delete Cascade) 이란 Child 페이지가 삭제될 경우 Parent 페이지의 링크도 같이 삭제하도록 한다.

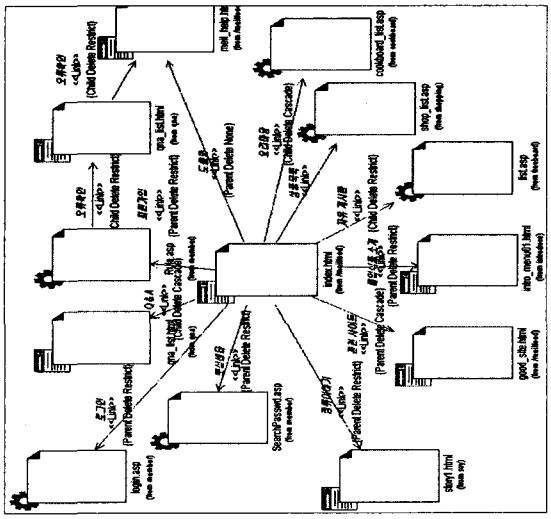
3. 웹 사이트에서 무결성 유지를 위한 WPMS 설계 및 구현

3.1 M사의 웹 사이트 개발시 확장 UML 적용

웹 페이지 문서들 사이에 존재하는 관계성(Relationship)과 제약조건을 정의하여 웹 페이지 문서의 삭제가 발생할 경우 웹 페이지의 무결성(Integrity)을 유지할 수 있도록 하는 WPMS(Web Page Integrity Management System)를 개발하였다.

WPMS는 확장 UML을 사용하며, 템플릿 코드를 생성한다. 본 논문에서는 웹 페이지간의 관계를 식별하고 제약조건을 정의하기 위한 확장 UML을 통한 웹 페이지 설계를 위하여 래쇼날 사의 래쇼날 로즈(Rational Rose)를 이용하였다.

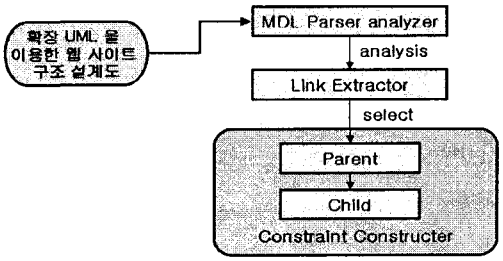
설계 단계에서는 웹 페이지간의 무결성 유지를 위한 관계 식별 및 정의를 하였고, 구현(코딩) 단계에서는 확장 UML을 통한 웹 페이지 설계를 쉽게 적용하기 위하여 WPMS을 개발하여 [그림 1]과 같은 결과를 얻을 수 있었다.



[그림 2] M사의 UML 웹 확장 구조도

3.2 WPMS 설계 및 코드 생성기 구현

WPMS 코드 생성기를 통해 Rational Rose의 소스를 분석하여 제약조건과 관계정의를 추출하고 Parent와 Child 페이지에 링크 식별 및 제약조건을 적용하는 처리 흐름도를 [그림 2]에서 기술하고 있다.



[그림 3] WPMS 처리 흐름도

3.2.1 MDL Parser Analyzer 모듈

설계된 내용을 프로그램 코딩 단계에서 사용이 용이토록 하기위하여 Rational Rose의 모델 파일 구조를 파악하여 ASP와 HTML 탬플릿 코드를 생성하여야 하지만 공식적으로 모델 파일이 문서화된 것은 없다. 다행히 모델 파일은 ASCII Code의 텍스트 파일 형태로 되어 있고, 그 구조가 중첩된 괄호들의 쌍으로 이루어진 트리 형태이기 때문에 코드 생성에 필요한 부분을 추출하여 분석할 수 있다.

3.2.2 링크 Extractor 모듈

링크 이름이 목록화된 것에서 코드 생성이 필요한 링크를 선택하면 Parent 페이지와 Child 페이지의 실

제 파일이 생성될 경로가 조합 생성되고 삽입되어져야 할 링크 이름과 제약조건이 화면에 출력된다.

3.2.3 Parent 페이지 생성 모듈

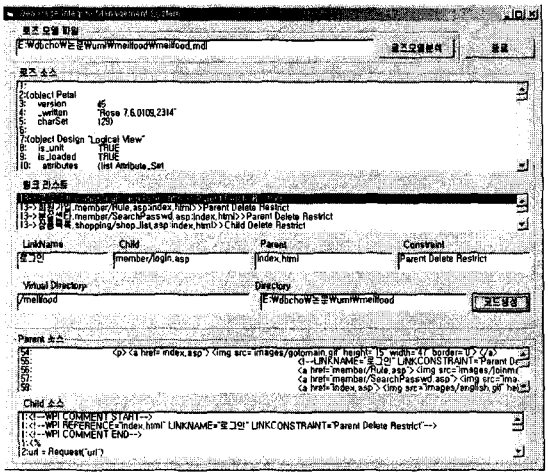
Parent 페이지에 해당하는 파일을 열어서 링크를 의미하고 있는 <A href> 태그를 찾아서 그 앞에 주석을 삽입하여 링크이름과 제약조건을 기술한다. 그런데, 템플릿 코드 생성 후 재생성을 할 경우 이미 그러한 내용이 삽입되어 있는지를 검사하고 없을 경우에만 기술하여야 한다.

3.2.4 Child 페이지 생성 모듈

START 줄과 END 줄 사이에 이 Child 페이지를 참조하고 있는 모든 Parent 페이지의 정보를 기술하는데, Parent 페이지 생성 모듈에서와 같이 템플릿 코드 생성 후 재생성을 할 경우 이미 그러한 내용이 삽입되어 있는 지를 검사하고 없을 경우에만 기술하여야 한다.

4. WPMS 수행 결과와 성능평가

코드 생성기를 통해 로즈 소스를 분석하여 제약조건과 관계정의를 추출하고 Parent와 Child 페이지에 링크 식별 및 제약조건을 적용하기 위하여 WPMS을 실행해야 한다. 확장 UML을 통해 생성된 Rational Rose 모델 파일을 분석하여 코드 생성을 하는 과정을 WPMS을 통해 구현되는 과정들을 전체적으로 [그림 3]에서 보여주고 있다.



[그림 4] WPMS 실행 화면

로그인 소스부분을 보면 확장 UML을 통해 생성된 MDL 파일을 읽어들여 소스를 생성한다. 링크 이름이 로그인이고 Parent 페이지와 Child 페이지의 이름과 제약조건에 관한 소스 부분을 보여주고 있다.

링크 리스트에서 '로그인' 부분을 선택했을 때 자동으로 실행되는 화면으로서 Child에는 Login.asp와 Parent의 Index.html, 그리고 Constraint의 Child Delete Restrict라는 제약조건이 나타난다.

4.1 웹 사이트 구성 비교

DLP의 발생에 관하여 비교 분석하기 위하여 M사의 웹 사이트에는 WPMS을 적용하여 웹 사이트를 구축하였고, IT사의 웹 사이트는 통상적인 개발 방법을 통하여 웹 사이트를 구축하였다. M사와 IT사 웹 사이트의 페이지 구성은 [표 2]와 같다.

[표 2] M사와 IT사 웹 사이트의 페이지 구성

구 분	M사 (www.meilfood.com)	IT사 (www.islandtour.co.kr)
정적 페이지 수(HTML)	45	80
동적 페이지 수(ASP)	61	132
링크 수	562	1,120
난이도*)	5.30	5.28

*) 난이도 = 링크 수 / 페이지 수(정적 페이지 수 + 동적 페이지 수)

4.2 웹 사이트 제작 공정

M사와 IT사의 웹 사이트 제작시 소요되는 일수, 투입인원, 일일 소요시간 등을 설계 및 코딩단계 그리고 테스트 과정은 [표 3,4]과 같이 나타났다.

[표 3] M사 웹 사이트 제작

구 분	설 계	코 딩	테 스 트	합 계
소요일수	7	28	5	40
투입인원	2	4	2	-
공수(일)	14	112	10	136

[표 4] IT사 웹 사이트 제작

구 분	설 계	코 딩	테 스 트	합 계
소요일수	7	45	5	57
투입인원	3	5	5	-
공수(일)	21	225	25	271

M사와 IT사 웹 사이트 사이의 페이지당 공수를 설계, 코딩, 테스트 단계로 나누어 알아본 결과를 [표 5]에서 보여주고 있다.

[표 5] M사와 IT사 웹 사이트간 페이지당 공수 비교

구 분	설 계	코 딩	테 스 트	합 계
M사	0.132	1.057	0.094	1.283*)
IT사	0.099	1.061	0.118	1.278**)

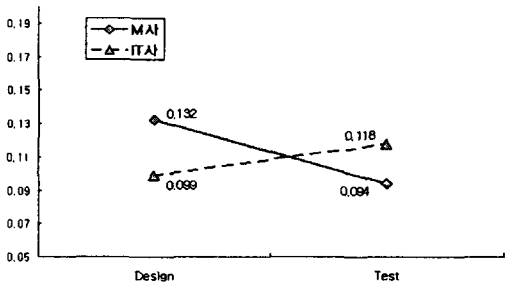
*) M사 : 전체 공수 합계(136) / 웹 페이지 수(106) = 1.283

**) IT사 : 전체 공수 합계(271) / 웹 페이지 수(212) = 1.278

M사와 IT사의 웹 사이트간 페이지당 공수를 비교한 내용은 [그림 4]와 같은데 여기에서 코딩의 경우 거의 차이가 없으므로 아래 그림에서는 생략하였다.

설계단계에서 0.132로 IT사의 0.099보다 페이지당 공수가 많이 투입되었으며 코딩 단계에서는 M사가 1.057의 페이지당 공수가 필요하였으며, IT사의 경우

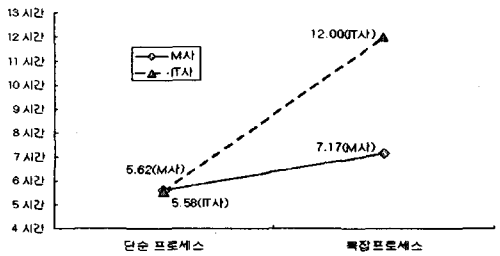
1.061의 페이지 공수가 각각 필요하였으나, 두 웹 사이트의 경우 코딩 단계에서는 별다른 공수의 차이를 보이지 않았다. 그러나 테스트 과정에서는 M사의 경우 0.094의 공수가 필요하였으며, IT사의 경우 0.118의 공수가 필요하였는데, 이는 IT사의 웹 사이트의 경우 설계에서 부족한 링크 정보 때문에 M사의 경우보다 테스트 과정에서 많은 공수가 투입됨을 알 수 있었다.



[그림 5] M사와 IT사 웹 사이트간 페이지당 공수 비교

4.3 M사와 IT사의 웹 사이트 유지보수 결과 분석

M사의 웹 사이트와 IT사의 웹 사이트를 개발한 후 유지보수 내용 중 단순 프로세스(링크를 2개이하 수정한 단순한 작업)와 복잡 프로세스(링크를 3개이상 수정한 다소 복잡한 작업)에 관하여 처리시간을 [그림 5]에서 보여주고 있다.



[그림 6] 두 웹 사이트간 단순, 복잡 프로세스 평균 처리시간 비교

두 웹 사이트간 단순, 복잡 프로세스의 평균처리 시간을 M사의 경우 초기 개발시 링크 식별명 및 제약조건의 기술에 따른 설계 단계에서 기존 개발방법보다 다소 시간이 많이 소요되었으나, 웹 페이지의 유지 보수시 웹 사이트의 링크 식별명 및 제약조건 등을 UML 확장 구조도를 통해 보면서 유지 보수를 함으로서 유지 보수에 따른 많은 시간을 단축하는 효과를 보여주었다.

5. 결 론

본 논문에서는 웹 페이지 문서들 사이에 존재하는 관계성과 제약 조건을 정의하고, 웹 사이트를 확장 UML을 통해 설계하고 설계된 내용을 웹 사이트 구현 단계에서 쉽게 적용하여 웹 사이트 무결성을 유지할 수 있도록 WPMS(Web Page Integrity Management System)를 개발하였다.

WPMS를 적용하여 제작한 두 웹 사이트의 성능 분석을 보면 코딩 단계에서는 별다른 공수의 차이를 보이지 않았다. 그러나 테스트 과정에서는 M사의 경우 0.094의 공수가 필요하였으며, IT사의 경우 0.118의 공수가 필요하였는데, 이는 IT사의 웹 사이트의 경우 설계에서 부족한 링크 정보 때문에 M사의 경우보다 테스트 과정에서 많은 공수가 투입됨을 알 수 있었다.

두 웹 사이트간 단순, 복잡 프로세스의 평균처리 시간을 보면, 단순 프로세스를 수정하는 유지 보수 건은 M사나 IT사 모두 5시간 정도를 나타냈으나, 복잡 프로세스에서 M사는 단순 프로세스보다 약간 높은 7.1 시간인데 반해 IT사의 경우 12시간의 유지보수시간이 소요되었다.

이와 같은 링크 식별명 및 제약조건을 설계단계에서부터 적용하여 웹 사이트를 구축하므로서 설계단계에서 기존의 웹 사이트 개발방법 보다 시간이 다소 지연되었으나, 테스트 및 유지보수 단계에서는 기존의 웹 개발 방법보다 시간 단축을 할 수 있었다.

참 고 문 헌

[1] 김수의, 장대용, "문서의 효율적인 검색을 위한 HTML 문서 변환 시스템", 한국정보과학회, 2000 가을 학술발표논문집(I), pp.184~186, 2000.

[2] Hugh C. Davis. "Referential Integrity of Links in Open Hypermedia Systems" in proceedings of ACM Hypertext '98, pittsburgh, PA, 207~216, June 1998.

[3] Hugh V. Davis, Wendy Hall, Ian Heath, Gary J. Hill, and Rob J. Wilkins. "Towards an Integrated Information Environment with Open Hypermedia Systems" in Proceedings of the ACM Conference on hypertext(ECH '92), Milano, Italy, 181~190, December 1992.

[4] Hugh davis. "Data Integrity Problems in an Open Hypermedia Link Service". PhD thesis, University of Southampton, 1995.

[5] Luc Moreau, nicholas gray. "A Community of Agents Maintaining Link Integrity in the World-Wide Web", University of Southampton, 1997.

[6] Helen Ashman, Hugh davis, "Missing th 404 : link integrity on the World Wide Web", Seventh International World Wide Web Conference, 1998.8.

[7] Stefano Ceri, G.Pelagatti, "Distributed Database Principles & Systems", McGraw-Hill, pp.61~62, 1984.