

XML을 이용한 자동차 정보 정비 시스템 구현

장식원*, 신용태
승실대학교 컴퓨터학과

An Implementation of Automobiles Maintenance System using XML

Shickwon Chang*, Yongtae Shin
Dept. of Computing, Soongsil University

요 약

XML은 이제 우리에게 아주 생소한 단어는 아니다. 어느 정도 익숙해졌고, 앞으로 웹이 XML을 포함할 것이라는 점을 의심하는 사람도 많지 않을 것이다. 그러나 실제로 XML이 어떻게 생성되고, 저장되며, 어떠한 방법으로 응용이 되는지에 대해서 아는 사람은 그리 많지 않다. XML은 자신만의 태그를 가질 수 있고, 그 태그에 대한 설명을 할 수 있으며, 스타일 지정 문서를 사용해서 하나의 문서를 여러 형태로 배포할 수 있다. 또한 XML은 구조적인 문서이기 때문에 보다 쉬운 관리, 검색 그리고 편집을 위해서 데이터베이스에 저장하는 것도 가능하다. 본 논문에서는 XML의 이러한 특성을 활용할 수 있는 웹 상에서 구현된 XML 편집기를 보여주고 XML 문서를 데이터베이스에 저장하는 방안을 제안한다. 그리고 활용의 예로써 자동차 정비정보 시스템 구축 사례를 소개한다.

1. 개 요

XML(eXtensible Markup Language)은 최근 인터넷 업계의 주요관심사로 부각되며 표준으로 자리 잡아가고 있다. 기존의 마크업 언어는 대표적으로 HTML(Hyper Text Markup Language)과 SGML(Standard Generalized Markup Language)이 있었다. HTML은 간단하여 많은 사용자를 인터넷으로 끌어들이는 장본인이다. 그러나 너무 단순하여 복잡한 응용 분야에서는 잘 적용하지 못하는 단점이 있다. 그에 비해 SGML은 표현능력은 뛰어나지만 문법들이 너무 복잡하여 사용하기 어려운 단점이 있다. 이 둘의 단점은 빼버리고 장점만을 취해서 만들어진 언어가 XML이다[1]. XML의 주요한 특징은 다음과 같다.

- 간단성
- SGML을 기본으로 함
- 확장성
- 문서의 재사용이 용이
- 일반화된 마크업, 복합 구조 지원

XML문서를 효율적으로 관리하기 위해서는 데이터베이스에 저장해야 한다. XML 문서는 구조적인 특성을 가지므로 쉽게 태그별로 나뉘어지고 저장된다. 본 논문에서는 그 저장기법을 소개한다.

또한 본 논문에서는 XML 문서 편집기를 제안한다. 이 편

집기는 통합 어플리케이션으로서 XML 문서의 저장에서부터 편집, 검색, 그리고 XSL 문서 편집을 지원한다.

XSL은 스타일 지정 문서로서 XML의 모양을 꾸미는 역할을 한다. XML의 다른 장점 하나는 XSL을 바꿔줌으로써 같은 내용의 문서라 할 지라도 새로운 형태의 문서로 보여지는 것이다. 이 XML 문서 편집기는 다양한 XSL을 수용할 수 있도록 하고 있다.

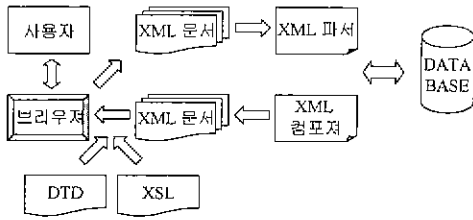
마지막으로 이 편집기를 이용한 자동차 정비 정보 시스템 구축사례를 설명한다.

구현은 Oracle server 8.0.5.0.0을 이용하였다.

2. XML 문서의 저장

XML 문서를 저장하기 위해서는 구조적 정보를 데이터베이스에 저장할 수 있어야 하고 이를 위해서는 태그별로 구분하여 저장하는 것이 문서의 검색과 관리를 훨씬 쉽게 해준다[1, 2]. [그림 1]은 XML 문서를 처리하는 과정을 나타낸 것이다.

파서는 입력된 XML 문서를 태그별로 분리하고 분리된 XML 문서의 구성요소들은 각각의 순서와 위치등의 정보와 함께 데이터베이스에 저장된다. 컴포저는 이 정보들을 종합하여 하나의 문서에 속하는 구성요소들을 불러들여 다시 XML 문서를 작성한다.



[그림 1] XML 문서 처리 과정

2.1 데이터베이스 테이블 구조

데이터베이스에는 각각의 구성요소를 저장하기 위한 항목이 테이블로 구성되어 있다. 테이블의 구조는 다음과 같다.

- ▶ company, car, ver, part, depart, work 테이블
각 항목을 저장하는 테이블이다[표 1].

[표 1] 항목 테이블

| num | name |
|--------|--------------|
| number | Varchar2, 40 |

- ▶ Title 테이블

문서가 가지고 있어야 하는 정보 중 서지정보 부분의 키 값을 가지고 있는 테이블이다[표 2]. num 값으로 문서번호를 정의하고 그 번호에 따르는 나머지 값으로 하나의 문서에 필요한 서지정보를 읽어온다.

[표 2] Title 테이블

| num | company | car | ver | part | depart | work |
|--------|---------|--------|--------|--------|--------|--------|
| number | number | number | number | number | number | number |

- ▶ Document 테이블

문서가 가지고 있어야 하는 정보중 본문에 해당하는 부분의 키 값을 가지고 있는 테이블이다[표 3]. num값에는 문서번호가 들어가고 content에는 본문에 있는 내용에 해당하는 키 값이 들어간다. 그리고 ordered에는 순서를 위한 번호가 들어가고 whered에는 그 값이 문장인지, 이미지인지 구분하는 값이 들어간다.

[표 3] Document 테이블

| num | content | ordered | whered |
|--------|---------|---------|--------|
| number | number | number | number |

- ▶ Phrase 테이블

본문에 해당하는 부분 중에서 문장에 해당하는 부분을 저장하는 부분이다[표 4]. content에 실제 문장이 들어간다.

[표 4] Phrase 테이블

| num | company | car | ver | part | depart | work | Content |
|--------|---------|--------|--------|--------|--------|--------|----------------|
| number | number | number | number | number | number | number | Varchar2, 1000 |

- ▶ Image 테이블

본문에 해당하는 부분 중에서 이미지에 해당하는 부분을 저장하는 부분이다[표 5]. content에 이미지의 파일 이름이 들어간다.

[표 5] Image 테이블

| num | company | car | ver | part | depart | work | Content |
|--------|---------|--------|--------|--------|--------|--------|----------------|
| number | number | number | number | number | number | number | Varchar2, 1000 |

2.2 XML 문서 저장의 예

위에서 정의한 테이블에 [표 6]의 문서를 저장한다. 이 문서를 파서에 넣어서 데이터베이스에 저장하게 되면 다음과 같이 각 테이블에 저장된다.

[표 6] XML 문서의 예

```
<company>현대</company>
<car>그랜저</car>
<ver>3.0</ver>
<part>엔진</part>
<depart>트랜스미션</depart>
<work>탈거</work>
<phrase>본네트를 연다.</phrase>
<phrase>엔진을 분해한다.</phrase>
<image>1_1</image>
<phrase>트랜스미션을 떼어낸다.</phrase>
<image>1_2</image>
```

[표 7] Company

| num | name |
|-----|------|
| 1 | 현대 |

[표 8] Car

| num | name |
|-----|------|
| 1 | 그랜저 |

[표 9] Ver

| num | name |
|-----|------|
| 1 | 3.0 |

[표 10] Part

| num | name |
|-----|------|
| 1 | 엔진 |

[표 11] Depart

| num | name |
|-----|-------|
| 1 | 트랜스미션 |

[표 12] Work

| num | name |
|-----|------|
| 1 | 탈거 |

[표 13] Title

| num | company | car | ver | part | depart | work |
|-----|---------|-----|-----|------|--------|------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

[표 14] Document

| num | content | ordered | whered |
|-----|---------|---------|--------|
| 1 | 1 | 1 | 1 |
| 1 | 2 | 2 | 1 |
| 1 | 1 | 3 | 2 |
| 1 | 3 | 4 | 1 |
| 1 | 2 | 5 | 2 |

[표 15] Phrase

| num | company | car | ver | part | depart | work | Content |
|-----|---------|-----|-----|------|--------|------|--------------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 본네트를 연다. |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 엔진을 분해한다. |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 트랜스미션을 떼어낸다. |

[표 16] Image

| num | company | car | ver | part | depart | work | Content |
|-----|---------|-----|-----|------|--------|------|---------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1_1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1_2 |

3. 구현

이 문서 편집기는 다음과 같은 구성요소를 가지고 있다.

- ▶ 입력 모듈 : DB에 XML 문서와 문서정보를 입력한다.
- ▶ 편집 모듈 : DB에 입력된 데이터를 새로운 데이터로 고치거나 삭제한다.
- ▶ 출력 모듈 : DB에 입력된 데이터를 XML 혹은 HTML로 보여준다.
- ▶ 검색 모듈 : 기존에 입력되어 있는 문서를 검색한다.
- ▶ XSL 편집기 : XSL 문서를 직접 작성한다.

3.1 입력 모듈

XML 입력기의 왼쪽에는 XML 문서를 검색하기 위한 서지정보를 입력하고, 오른쪽 입력창에는 XML 문서의 본문을 입력한다.

입력이 완료되면 파서는 입력된 본문을 해석한다. 그리고 태그별로 데이터 베이스에 저장한다. 나중에 문서를 재구성할 때 필요한 정보들(순서, 문서이름등)도 같이 저장된다.

3.2 편집 모듈

XML 편집기는 DB에 저장한 XML 문서를 필요에 따라 수정 및 보완하기 위한 기능으로 중앙에 위치한 화면의 양식에 따라 편집하고자 하는 문서의 서지정보를 선택하고 확인을 누르면 거기에 해당하는 문서를 보여준다.

편집창에 수정 및 보완 할 문서가 보여지면 문서의 원하는 부분에서 새로 꾸미거나 수정할 수 있다. 또한 필요 없는 부분은 삭제할 수 있다.

3.3 출력 모듈

XML 출력기는 출력하기 위한 XML문서의 서지정보를 중앙에 위치한 양식에 맞추어 선택해주면 거기에 해당하는 문서를 XML 혹은 HTML의 모습으로 볼 수 있다. XML과 HTML의 선택은 화면 중앙의 라디오 버튼을 통해 이루어지며, HTML을 지원하는 이유는 아직은 익스플로러 5.0 에서만 XML을 지원하기 때문이다.

그리고 XML의 장점인 다양한 문서 양식을 지원하기 위해 하단에 위치한 입력창에 원하는 XSL 파일의 이름을 적을 수 있다. 다양한 XSL을 지원하는 이유는 XSL을 변경함으로써 같은 문서를 가지고 다양하게 표현할 수가 있기 때문이다 만약 적지 않는다면 본 시스템에서 지원하는 기본 XSL 파일명이 들어가게 된다.

3.4 검색 모듈

검색은 구분검색과 이미지 검색을 할 수 있으며, 검색 창 중앙에 위치한 양식에 따라 값을 선택하고 확인 버튼을 누르면 검색하고자 하는 문서의 본문이나 이미지를 볼 수 있다.

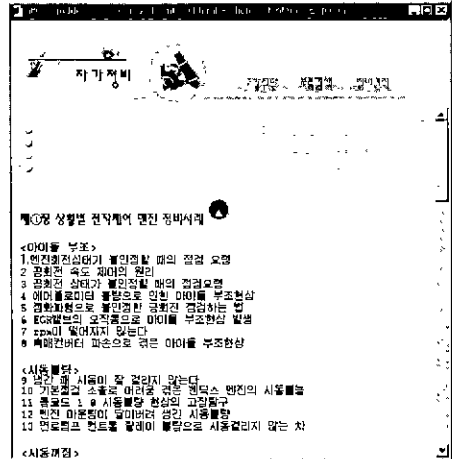
구분검색의 경우 검색 창에 XML 문서를 작성하거나 편집할 때 자신이 원하는 본문을 포함하는 문서를 검색해 보여주며, 본문의 내

용을 복사하여 문서 작성에 사용함으로써 문서 작성 작업을 보다 쉽게 할 수 있도록 해준다.

이미지검색의 경우 검색 결과로 파일이름과 이미지를 보여준다.

3.5 구현 사례

[그림 2]는 구현된 자동차 정보 정비 시스템이다



[그림 2] 자동차 정보 정비 시스템

4. 결론 및 향후 과제

XML의 수요가 계속 증가함에 따라 이를 저장하고 활용할 수 있는 어플리케이션은 계속 개발될 것이다. 이 논문에서는 이러한 어플리케이션의 예로 자동차 정비 정보 시스템을 제시하였다. 이 시스템은 꼭 자동차 정비에만 국한되지 않고 데이터 베이스와 태그를 약간 수정해 주면 어느 분야에서라도 사용될 수 있을 것이다.

향후과제로는 문서 저장방안의 효율성을 검증하고, 좀 더 완벽한 파싱을 할 수 있는 파서를 개발하고, 그리고 다양한 분야에서 사용될 수 있도록 범용화 하는 것이다.

5. 참고문헌

- [1] 류진영, 김찬홍, 유영호, 김경석, "관계형 데이터베이스에서 XML 문서 저장 방안 연구," 한국정보과학회 가을 학술발표논문집 Vol. 25, no. 2 1998.
- [2] 김규태, 현득창, 이수연, 정광철, "관계형 데이터베이스를 이용한 SGML 문서 처리," 정보과학회논문지(C) 제 3권 제3호 1997
- [3] Tim Bray, Jean Paoli, C. M. "Extensible Markup Language(XML)," <http://www.w3.org/TR/WD-xml-lan-g.html>.
- [4] 이희주, 장재우, 심부성, 주종철, "구조화 문서를 위한 정보 검색 인덱스의 설계," 한국정보과학회 학술발표논문집 Vol. 24. no. 2 1997.
- [5] Jon Bosak, "XML, Java, and the future of the Web", <http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm>.
- [6] 이용배, 손기락 "SGML 문서 저장을 위한 스키마 생성기 및 자동삽입기의 설계 및 구현," 정보과학회 논문지 1997.
- [7] Tim Bray, "Lark XML parser," <http://www.textuality.com/Lark/>.