

CORBA 기반의 XML 정보 관리 시스템 설계

문수룡*, 이정민, 김광중, 이연식
군산대학교 컴퓨터정보과학과
e-mail:kkjkim@kunsan.ac.kr

Design of XML Information Management System based on CORBA

Suryong Moon*, Jeongmin Lee, Kwangjong Kim, Yonsik Lee
Dept. of Computer Information Science, Kunsan National University

요 약

본 논문은 웹 상에서의 데이터 교환 및 정보 전달은 물론 분산 환경 내에 산재한 어플리케이션들 간의 통신 및 정보 시스템들 간의 상호 운용을 지원하는 CORBA 기반의 XML 정보 관리 시스템을 설계한다. 최근 웹 상에서 정보교환을 통해 다양한 업무들을 효율적으로 처리하려 함에 따라 분산 환경에서의 컴퍼넌트 기반과 웹 기반 어플리케이션들의 연동이 요구된다. 그러나, 이러한 어플리케이션들의 연동 시 직접적인 호환성이 결여되어 별도의 프로토콜 지원이 필요하다. 따라서, 본 논문에서는 이러한 호환성 문제를 해결하기 위해 XML을 사용하여 플랫폼 및 프로그래밍 언어에 대한 독립성과 확장성을 제공하고, 객체 직렬화를 통해 직렬화된 XML 문서를 CORBA의 IIOP를 이용해 전송함으로써 데이터 교환 및 공유가 가능하도록 지원하는 XML 정보 관리 시스템을 설계한다. 제안된 시스템은 웹 상에서의 정보 교환 및 공유를 지원함으로써 보다 유연한 정보 교류 환경을 제공하고, 이질적인 어플리케이션들로 구성된 비즈니스 로직들 사이의 정보 처리에 있어 상호 운용성을 지원하여 효율적인 업무 처리가 가능하도록 한다. 또한, XML2ORDB 저장기와 ORDB2XML 생성기를 구성 모듈로 설계하여 웹 상에서의 모든 정보를 공통 포맷인 XML 형태로 저장 및 관리함으로써 정보의 재사용성 및 효율적인 관리를 유도한다.

1. 서론

오늘날 인터넷의 급속한 확산으로 가상 공간에서의 전자 상거래는 물론 많은 인터넷 프로그램들이 구축되어 사용자에게 다양한 정보들을 제공하고 있다. 하지만 최근 이러한 웹 어플리케이션들의 구축에 있어 가상공간에서의 일방적인 정보 제공 목적이 아닌 서로 간의 정보공유와 데이터 교환에 초점이 맞춰지고 있다. 이는 대부분의 사람들 사이에 잘 알려진 넷스케이프나 익스플로러와 같은 일반적인 웹 브라우저만으로도 서비스를 이용할 수 있어 다른 프로그램의 설치가 불필요하고 플랫폼에 독립적으로 정보를 이용할 수 있다는 장점을 제공하기 때문이다. 최근 기업들도 초고속 정보통신망의 구축이 활발히 진행됨에 따라 이를 이용한 웹상에서의 정보교환을 통해 업무의 효율화를 이루려는 노력이 점차 활성화 되고 있다. 그러나, 기존 분산 환경에서 어플리케이션간 통신 수단으로써 웹을 이용하려할 경우, 컴퍼넌트 기반의 어플리케이션들과 웹의 직접적인 호환성이 결여되어 별도의 프로토콜 지원이 요구된다[3,4,7]. 이에 따라 컴퍼넌트 기반의 응용 프로그램들과 웹과의 호환성 문제 해결을 위해 공통 데이터 교환 포맷인 XML의 사용이 증가하고 있으며, 또한 기존 분산 환경에서 이종의 네트워크 및 객체간 통신, 네트워크 트래픽 분산, 서버의 과부하 방지, 상호운용성 등을 지원하는 CORBA와 같은 미들웨어와 XML과의

연동이 요구되고 있다[5,6,7]. 따라서, 본 논문에서는 웹상에서의 데이터 교환 및 정보 전달은 물론 분산 환경내에 산재한 서로 다른 응용 프로그램 간의 통신 및 정보 시스템들 간의 상호 운용을 지원하는 CORBA 기반의 XML 정보관리 시스템을 설계한다. 제안된 시스템에서는 효율적인 XML 문서 저장 및 관리를 위해 DTD에 기반한 새로운 데이터베이스 스키마 생성 방법을 제시하고, 데이터베이스에 XML 문서를 저장하기 위한 XML2ORDB 저장기 및 역으로 데이터베이스의 데이터들로부터 XML 문서를 생성하기 위한 ORDB2XML 생성기를 개발함으로써 보다 효과적인 XML 문서의 저장 및 관리를 수행할 수 있도록 지원한다.

본 논문은 2장에서 XML 문서를 데이터베이스에 저장하는 방법과 분산환경에서의 XML 활용 예로서 SOAP(Simple Object Access Protocol)과 SCOAP(Simple CORBA Object Access Protocol)에 대해 설명하고, 3장에서는 XML DTD 및 DTD를 통한 데이터베이스 스키마 생성, CORBA 상에서의 XML 문서 교환을 위한 인터페이스 및 CORBA 객체 생성에 대해 기술한다. 4장은 분산환경내에서 CORBA와 XML 데이터 모델을 이용하여 원격지에 존재하는 시스템 간에 XML 문서를 교환 및 공유하기 위한 XML 정보관리 시스템을 설계한다. 마지막 5장에서는 결론 및 향후 연구과제에 대해 기술한다.

2. 관련 연구

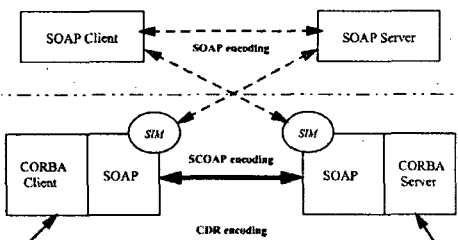
2.1 XML 문서의 데이터베이스 저장 방법

XML 문서를 데이터베이스에 저장하는 방법은, 저장 기법에 따라 XML 문서 전체를 한번에 저장하는 비분할 모델과, 문서를 엘리먼트 단위로 나누어 저장하는 분할 모델이 있다. 비분할 모델은 XML 문서의 전체 내용을 하나의 CLOB으로 데이터베이스에 저장하는 방식으로, 문서 내의 엘리먼트 태그들의 구조 및 위치정보를 저장하기 위한 엘리먼트 테이블과 문서 전체를 저장하기 위한 문서 테이블로 구성된다[1, 8]. 이러한 비분할 저장 방법은 DTD와 독립적인 데이터 모델을 사용함으로써 생성되는 테이블의 수와 데이터의 중복이 적고 검색 효율이 우수한 특성을 가지나, XML 문서에 엘리먼트의 갱신이 발생할 경우, 다른 엘리먼트들의 위치 정보가 모두 수정되어야 함으로써 오버헤드 및 데이터베이스 일관성 유지 문제가 있다.

분할 모델은 문서의 내용을 DTD를 기준으로 엘리먼트와 속성으로 나누어 저장하는 방식으로, 문서의 실제 내용을 가진 각각의 단말 엘리먼트 안에 문서의 내용이 분할되어 저장된다[2, 8]. 특정 엘리먼트의 갱신 시 이 엘리먼트와 관련된 부모, 자식, 전후 엘리먼트의 정보만을 변경해 주면 뒤으로 비분할 모델보다 쉽게 문서 내용을 갱신할 수 있어 문서의 버전 관리가 쉽고 동일한 내용을 갖는 노드들을 공유할 수 있다. 또한, 검색의 경우에 있어 검색 대상 엘리먼트가 다른 엘리먼트를 포함하는 상위 엘리먼트라면, 그 하위 엘리먼트를 모두 포함하기 위해 각각의 단말 엘리먼트를 순회하여 결과를 구성해야하므로 검색 과정이 복잡하고 많은 시간이 소요되는 단점을 가진다.

2.2 분산환경에서의 XML 활용

SOAP(Simple Object Access Protocol)은 XML을 사용하여 데이터와 명령을 확장 가능한 방법으로 표시하기 위한 산업 표준으로서, 분산 환경에서 정보를 주고받을 수 있도록 고안된 아주 가벼운 프로토콜이다[3,4]. SOAP은 HTTP(Hypertext Transfer Protocol)를 사용하여, 인터넷에 텍스트 명령어를 보내기 위해 XML 구문을 사용함으로써 기존의 인터넷 인프라스트럭처(Infrastructure)를 사용하여 응용 프로그램이 방화벽에 의해 뜻하지 않게 블록되는 일이 없이 다른 응용 프로그램과 직접 통신할 수 있게 만드는 방법이다. 한편, CORBA 기반의 시스템 아키텍처 상에서 이러한 SOAP을 사용하여 웹과의 상호연성을 지원하기 위해 SCOAP(Simple CORBA Object Access Protocol)이 표준으로 제안되었다[5,6]. SCOAP은 CORBA와 SOAP에 기반한 시스템 상에서 서로 간의 인터랙션을 지원하기 위해 고안된 프로토콜로서, CORBA+SOAP 클라이언트와 CORBA+SOAP 서버 사이에서 SCOAP 인코딩을 통한 인터랙션에 있어 SOAP/HTTP로 IIOP의 터널링 효과를 가능하게 하고, IDLtoXMLSchema 매핑을 필요로 하게 된다. 다음 [그림 1]은 CORBA와 SOAP 기반 시스템에서의 상호연용성을 나타낸다.



[그림 1] CORBA와 SOAP 기반 시스템에서의 상호연용성

3. XML 데이터 모델과 데이터베이스 스키마

3.1 XML 데이터 모델

XML DTD는 XML 문서의 데이터 모델을 위한 논리적인 구조 정보를 표현하는 것으로 XML 문서가 어떻게 구조화되어야 하고 어떤 요소들이 포함되어야 하며 어떤 종류의 데이터가 포함되어야 하는지 등을 규정하는 규칙을 정의하고 있다. XML DTD는 특히 별개의 서로 다른 비즈니스 파트너 사이에 신뢰할 수 있는 데이터 교환을 가능하게 한다. 이는 DTD가 XML 문서에 대한 기술(description)을 가능하게 하여 표준화된 유효성 검증 방법을 제공하기 때문이다. 즉, 하나의 DTD를 동일한 업무를 수행하는 비즈니스 파트너들 간에 공유함으로써 DTD의 구조에 맞게 생성되는 서로 다른 비즈니스 파트너들의 XML 문서들은 DTD와 유효성 검증 과서에 의해 유효한 문서로 검증되어 서로 간에 교환이 가능하게 된다. 다음 [그림 2]는 XML DTD의 예로서 학생정보에 관한 데이터 모델을 표현하는 DTD의 생성 예이다.

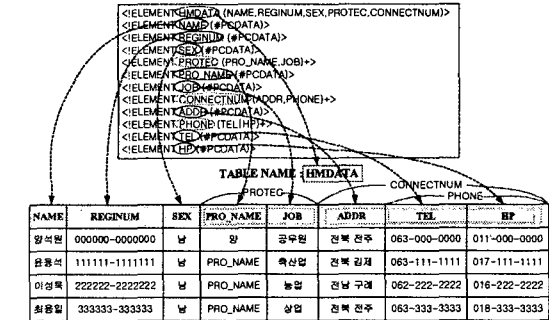
```

<ELEMENT NAME (#PCDATA) >
<ELEMENT REGNUM (#PCDATA) >
<ELEMENT SEX (#PCDATA) >
<ELEMENT PRO_NAME (#PCDATA) >
<ELEMENT JOB (#PCDATA) >
<ELEMENT ADDR (#PCDATA) >
<ELEMENT TEL (#PCDATA) >
<ELEMENT ZIP (#PCDATA) >
<ELEMENT EIP (#PCDATA) >
<ELEMENT SPECIAL (#PCDATA) >
<ELEMENT USRN (#PCDATA) >
<ELEMENT USRN2 (#PCDATA) >
<ELEMENT USRN3 (#PCDATA) >
<ELEMENT USRN4 (#PCDATA) >
<ELEMENT USRN5 (#PCDATA) >
<ELEMENT USRN6 (#PCDATA) >
<ELEMENT USRN7 (#PCDATA) >
<ELEMENT USRN8 (#PCDATA) >
<ELEMENT USRN9 (#PCDATA) >
<ELEMENT USRN10 (#PCDATA) >
<ELEMENT USRN11 (#PCDATA) >
<ELEMENT USRN12 (#PCDATA) >
<ELEMENT USRN13 (#PCDATA) >
<ELEMENT USRN14 (#PCDATA) >
<ELEMENT USRN15 (#PCDATA) >
<ELEMENT USRN16 (#PCDATA) >
<ELEMENT USRN17 (#PCDATA) >
<ELEMENT USRN18 (#PCDATA) >
<ELEMENT USRN19 (#PCDATA) >
<ELEMENT USRN20 (#PCDATA) >
<ELEMENT USRN21 (#PCDATA) >
<ELEMENT USRN22 (#PCDATA) >
<ELEMENT USRN23 (#PCDATA) >
<ELEMENT USRN24 (#PCDATA) >
<ELEMENT USRN25 (#PCDATA) >
<ELEMENT USRN26 (#PCDATA) >
<ELEMENT USRN27 (#PCDATA) >
<ELEMENT USRN28 (#PCDATA) >
<ELEMENT USRN29 (#PCDATA) >
<ELEMENT USRN30 (#PCDATA) >
<ELEMENT USRN31 (#PCDATA) >
<ELEMENT USRN32 (#PCDATA) >
<ELEMENT USRN33 (#PCDATA) >
<ELEMENT USRN34 (#PCDATA) >
<ELEMENT USRN35 (#PCDATA) >
<ELEMENT USRN36 (#PCDATA) >
<ELEMENT USRN37 (#PCDATA) >
<ELEMENT USRN38 (#PCDATA) >
<ELEMENT USRN39 (#PCDATA) >
<ELEMENT USRN40 (#PCDATA) >
<ELEMENT USRN41 (#PCDATA) >
<ELEMENT USRN42 (#PCDATA) >
<ELEMENT USRN43 (#PCDATA) >
<ELEMENT USRN44 (#PCDATA) >
<ELEMENT USRN45 (#PCDATA) >
<ELEMENT USRN46 (#PCDATA) >
<ELEMENT USRN47 (#PCDATA) >
<ELEMENT USRN48 (#PCDATA) >
<ELEMENT USRN49 (#PCDATA) >
<ELEMENT USRN50 (#PCDATA) >
<ELEMENT USRN51 (#PCDATA) >
<ELEMENT USRN52 (#PCDATA) >
<ELEMENT USRN53 (#PCDATA) >
<ELEMENT USRN54 (#PCDATA) >
<ELEMENT USRN55 (#PCDATA) >
<ELEMENT USRN56 (#PCDATA) >
<ELEMENT USRN57 (#PCDATA) >
<ELEMENT USRN58 (#PCDATA) >
<ELEMENT USRN59 (#PCDATA) >
<ELEMENT USRN60 (#PCDATA) >
<ELEMENT USRN61 (#PCDATA) >
<ELEMENT USRN62 (#PCDATA) >
<ELEMENT USRN63 (#PCDATA) >
<ELEMENT USRN64 (#PCDATA) >
<ELEMENT USRN65 (#PCDATA) >
<ELEMENT USRN66 (#PCDATA) >
<ELEMENT USRN67 (#PCDATA) >
<ELEMENT USRN68 (#PCDATA) >
<ELEMENT USRN69 (#PCDATA) >
<ELEMENT USRN70 (#PCDATA) >
<ELEMENT USRN71 (#PCDATA) >
<ELEMENT USRN72 (#PCDATA) >
<ELEMENT USRN73 (#PCDATA) >
<ELEMENT USRN74 (#PCDATA) >
<ELEMENT USRN75 (#PCDATA) >
<ELEMENT USRN76 (#PCDATA) >
<ELEMENT USRN77 (#PCDATA) >
<ELEMENT USRN78 (#PCDATA) >
<ELEMENT USRN79 (#PCDATA) >
<ELEMENT USRN80 (#PCDATA) >
<ELEMENT USRN81 (#PCDATA) >
<ELEMENT USRN82 (#PCDATA) >
<ELEMENT USRN83 (#PCDATA) >
<ELEMENT USRN84 (#PCDATA) >
<ELEMENT USRN85 (#PCDATA) >
<ELEMENT USRN86 (#PCDATA) >
<ELEMENT USRN87 (#PCDATA) >
<ELEMENT USRN88 (#PCDATA) >
<ELEMENT USRN89 (#PCDATA) >
<ELEMENT USRN90 (#PCDATA) >
<ELEMENT USRN91 (#PCDATA) >
<ELEMENT USRN92 (#PCDATA) >
<ELEMENT USRN93 (#PCDATA) >
<ELEMENT USRN94 (#PCDATA) >
<ELEMENT USRN95 (#PCDATA) >
<ELEMENT USRN96 (#PCDATA) >
<ELEMENT USRN97 (#PCDATA) >
<ELEMENT USRN98 (#PCDATA) >
<ELEMENT USRN99 (#PCDATA) >
<ELEMENT USRN100 (#PCDATA) >
    
```

[그림 2] 학생정보 DTD 생성 예

3.2 XML DTD 기반의 데이터베이스 스키마 생성

XML DTD가 가진 구조 및 관계 정보를 기반으로 관계형 데이터베이스 스키마, 즉 관계형 테이블로 사상할 수 있도록 XML 문서를 저장 및 관리할 수 있다. XML DTD를 관계형 테이블로 사상시키는 방법으로는 문서의 과도한 단편화 문제 및 엘리먼트에 따른 테이블 생성 문제, 중복 데이터의 저장 등과 같은 문제점들을 가짐으로써, 본 논문에서는 DTD를 이용한 새로운 데이터베이스 스키마 생성 방법을 제안한다. 본 논문에서 제안하는 DTD를 통한 관계형 데이터베이스 스키마 생성 방법은 기존 관계형 데이터 모델이 2차원 테이블로 트리구조 형태를 가진 XML 문서를 저장할 때 발생하는 노드들의 깊이문제를 해결할 수 있는 모델이다. 다음 [그림 3]은 XML DTD에 대한 관계형 데이터베이스 스키마이다.

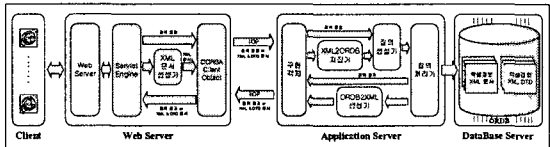


[그림 3] XML DTD에 대한 데이터베이스 스키마 생성

4. XML 정보 관리 시스템 설계

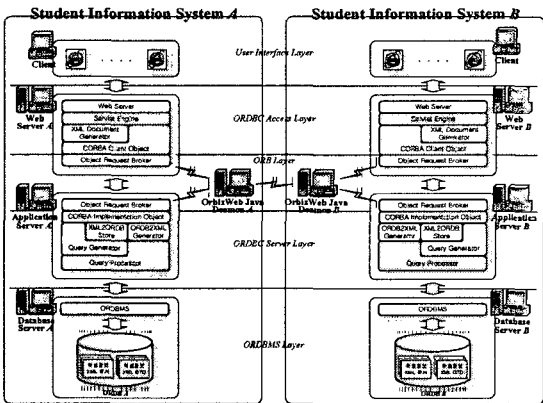
4.1 XML 정보 관리 시스템

XML 정보 관리 시스템은 분산 환경내에서 CORBA와 XML 데이터 모델을 이용하여 원격지에 존재하는 시스템 간에 XML 문서를 교환 및 공유하기 위한 시스템이다. 전체 시스템 구조는 사용자 인터페이스 층과 ORDBC (Object-Relational DataBase Connectivity) 접근 층, ORB 층, ORDBC서버층, ORDBMS(Object-Relational DataBase Management System)층으로 구성된다. 다음 [그림 4]는 XML 정보를 관리하기 위한 CORBA 기반의 XML 정보 관리 시스템에 대한 전체 구성도이다.



[그림 4] XML 정보 관리 시스템의 전체 구성도

CORBA 기반의 XML 정보 관리 시스템은 아래 [그림 5]와 같이 분산 환경내에서 원격지의 두 시스템 간에 XML 정보를 서로 교환 및 공유할 수 있도록 지원한다. 실제 두 그룹 간의 연결은 CORBA 서버 객체에 의해 등록된 구현 객체들에 대한 프록시 객체를 유지 및 관리 기능을 수행하는 OrbixWeb Java Deamon이나 VisiBroker의 OSAgent를 통해 서로 다른 도메인내의 CORBA 클라이언트 객체와 CORBA 구현 객체 간의 통신이 이루어진다. CORBA를 사용하여 기존의 각 도메인내에서 사용하던 이종의 비즈니스 로직들 간의 통신을 유지하고 서로 간에 공통된 데이터 포맷인 XML 문서를 교환 및 공유함으로써 호환성을 제공할 수 있다. 다음 [그림 5]는 두 도메인 내 각 XML 정보 시스템들 간의 통신과정을 나타낸다.



[그림 5] 두 도메인내 XML 정보 관리 시스템 간의 통신

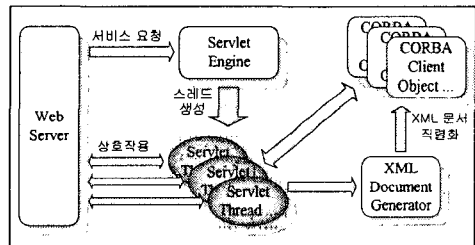
4.2 사용자 인터페이스 층

XML 정보 관리 시스템의 사용자 인터페이스는 시스템 특성상 빈번한 데이터 갱신이 이루어짐으로써 사용자들의 다양한 서비스를 효율적으로 처리하기 위한 모듈로 구성한다. 이는 다양한 각 사용자들의 요구를 효율적으로 처리하기 위해 자바의 스텔드를 사용하여 웹 서버의 부하를 줄일 수 있도록 JSP 페이지로 인터페이스를 구성하고, XML 문서 형태의 정보를 보다 쉽게 관리하기 위해 학생 정보 DTD에서의 학적사항, 인적사항, 출결사항, 교과발달

상황 등과 같은 루트 엘리먼트의 자식 엘리먼트들에 대한 메뉴들로 구성한다. 또한, 각 학생들의 학적정보에 대한 가시성을 높이고 각 학생들의 정보 조작을 위한 메뉴기능을 쉽게 하기 위해 트리형태로 학생 리스트를 구성한다.

4.3 ORDBC 접근 층

ORDBC 접근 모듈은 이질적 환경에서의 통신을 가능하게 하는 미들웨어인 CORBA의 클라이언트 객체들을 포함하고, ORDBC 서버 모듈의 CORBA 구현 객체와 통신을 수행하여 실제 클라이언트 사용자의 요구를 처리한다. 또한, 사용자의 요구로부터 XML 문서를 생성하여 미들웨어를 통해 XML 문서를 프로토콜 형식으로 송신/수신이 가능하게 함으로써 XML과 CORBA와의 연동이 이루어지게 한다. 즉, XML 문서 생성기를 통해 생성된 사용자의 요구에 대한 XML 문서를 CORBA 클라이언트 객체내에서 객체 직렬화를 통해 CORBA 구현 객체에 전송하게 되면, CORBA 구현 객체는 이를 원래 상태의 XML 문서로 복원하게 된다. 한편, CORBA의 상호운용성을 기반으로 CORBA 클라이언트 객체와 다른 도메인 내의 서버 구현 객체를 연결함으로써 서로 다른 도메인 간에 XML 문서를 교환 및 공유할 수 있다. 다음 [그림 6]은 ORDBC 접근 모듈의 구조 및 동작과정을 나타낸다.



[그림 6] ORDBC 접근 모듈의 구조 및 동작과정

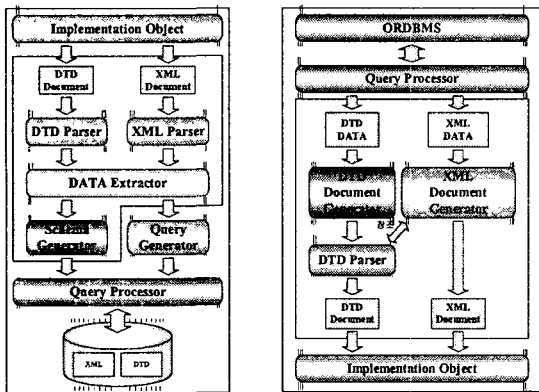
4.4 ORDBC 서버 층

4.4.1 XML2ORDB 저장기와 ORDB2XML 생성기

XML2ORDB 저장기는 XML과 XML DTD 문서를 객체관계형 데이터베이스에 저장하는 모듈로서, DTD 파서, XML 파서, 데이터 추출기, 스키마 생성기로 구성된다. DTD 파서는 CORBA 구현 객체로부터 DTD 문서를 넘겨받아 파싱을 통해 DTD 파스트리를 생성하여 트리의 각 노드 정보를 추출할 수 있도록 지원한다. XML 파서는 XML 문서를 파싱하여 XML 파스트리를 생성하고, 생성된 트리로부터 XML 문서에서의 단말 엘리먼트의 값, 즉 PCDATA를 추출할 수 있도록 한다. 이때, XML 파서의 파싱은 DTD 문서의 파싱 및 데이터 추출, 스키마 생성기를 통해 데이터베이스 스키마가 ORDB에 생성된 후 수행된다. 데이터 추출기는 파싱을 통해 생성된 각각의 DTD 파스트리와 XML 파스트리로부터 ELEMENT, ATTLIST, ENTITY, Content-Category, Content Model 등과 XML 문서에서의 단말 엘리먼트가 가진 PCDATA 값을 추출하여 스키마 생성기와 질의 생성기가 해당 질의를 생성할 수 있도록 지원한다. 스키마 생성기는 데이터 추출기를 통해 추출된 DTD에 대한 요소 정보들을 이용하여 XML 문서를 저장하기 위한 데이터베이스 스키마 생성 질의를 만드는 모듈이다.

ORDB2XML 생성기는 사용자의 요구에 대해 ORDB로부터 데이터 검색을 통해 얻은 질의 결과를 이용하여 XML 문서와 XML DTD 문서를 생성하는 모듈로서, DTD 문서 생성기와 DTD 파서, XML 문서 생성기로 구

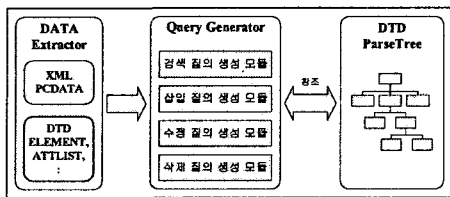
성된다. DTD 문서 생성기는 사용자의 요구에 대한 질의 결과로부터 이에 해당하는 XML 문서를 생성하려 할 경우, 이러한 XML 문서의 구조 정보를 표현하는 DTD 문서를 생성하기 위한 모듈이다. DTD 파서는 DTD 문서 생성기로부터 XML DTD 문서를 넘겨 받아 파싱을 통해 메모리 상에 DTD 파스트리를 생성함으로써 XML 문서 생성기가 질의 처리기에 의해 넘겨진 질의 결과로부터 DTD 파스트리를 참조하여 XML 문서를 생성하도록 지원한다. XML 문서 생성기는 사용자의 요구에 대한 질의 결과로부터 DTD 문서의 파싱을 통해 생성한 DTD 파스트리를 참조하여 XML 문서를 생성하는 모듈이다. 다음 [그림 7] (a) 와 (b)는 각각 XML2ORDB 저장기와 ORDB2XML 생성기의 구성도이다.



[그림 7] (a) XML2ORDB 저장기 구성도
(b) ORDB2XML 생성기 구성도

4.4.2 질의 생성기와 질의 처리기

질의 생성기는 데이터베이스의 데이터를 검색, 삽입, 수정, 삭제하기 위한 질의들을 생성하는 모듈이다. 질의 생성기는 데이터 추출기로부터 XML 문서의 PCDATA 값과 DTD 문서의 ELEMENT, ATTLIST, ENTITY 등과 같은 데이터를 이용하여 메모리 상에 로드된 DTD 파스트리를 참조하여 데이터를 조작할 수 있는 질의를 생성한다. 다음 [그림 8]은 질의 생성기에 대한 구조를 나타낸다.



[그림 8] 질의 생성기의 구조

질의 처리기는 데이터베이스에 대한 데이터 연산 수행 전, XML2ORDB 문서 저장기의 스키마 생성기와 질의 생성기로부터 스트링 형태의 질의를 입력받아 타당성을 검사하고, 데이터베이스 서버의 ORDBMS를 연결하여 해당 질의를 처리한다. 질의 처리기와 데이터베이스의 연결은 내부에 포함된 데이터베이스 연동 모듈을 통해 JDBC 드라이버를 이용하여 연결하며, 질의처리기는 질의에 대한 결과를 데이터베이스로부터 받아 ORDB2XML 문서 생성기의 입력으로 넘겨줌으로써 DTD 및 XML 문서를 생성할 수 있도록 지원한다.

4.5 ORDBMS 층

ORDBMS 층은 실제 백엔드 데이터베이스 시스템으로서 데이터베이스 시스템을 구성하는 DBMS와 데이터베이스로 이루어져 있다. 본 시스템을 구성하는 백엔드 데이터베이스는 ORACLE 8i로서 ORDBMS는 질의 처리기와 연결되어 요구된 데이터들을 데이터베이스로부터 추출하여 이를 결과로 반환하는 역할을 수행한다. 또한, ORDBMS는 데이터베이스 내의 데이터 일관성 유지, 데이터 및 메타데이터 추출, 무결성 제어 등과 같은 기능을 수행한다.

5. 결론 및 향후 연구과제

본 논문에서는 웹 상에서의 모든 정보를 공통 데이터 포맷인 XML 문서 형태로 관리하고, 서로 간의 정보 교환 및 정보 공유를 지원하는 CORBA 기반의 XML 정보관리 시스템을 설계하였다. 제안된 시스템은 분산 환경 내에 산재한 서로 다른 응용 프로그램들과 웹과의 호환성 문제를 해결하기 위해 CORBA 기반의 시스템 구조에서 XML을 사용하여 플랫폼 및 프로그래밍 언어에 대한 독립성과 확장성을 확보할 수 있도록 하며, 객체 직렬화를 통해 직렬화된 XML 문서를 CORBA의 IIOP를 이용해 전송함으로써 데이터 교환 및 공유가 가능하도록 한다. 이를 통해 기업에서의 다양한 업무 처리 시 보다 유연한 정보 교류 환경을 제공하고, 이질적인 기업 업무환경에서 서로 다른 비즈니스 로직들 간의 정보 처리에 있어 상호 운용성을 지원하여 효율적인 업무 처리가 가능하도록 한다. 또한, 설계된 시스템에서는 웹 상에서의 모든 정보를 XML 형태로 저장 및 관리하기 위하여, 데이터베이스에 XML 문서 형태의 정보를 저장하는 XML2ORDB 저장기와 데이터베이스로부터 XML 문서를 자동으로 생성하는 ORDB2XML 생성기를 구성하여 정보의 재사용성을 높이고 보다 효율적인 정보 관리가 이루어지도록 지원한다.

향후에는 설계된 시스템의 구현 및 실제 적용에 있어서 발생할 수 있는 문제점들에 대한 분석과 해결책들에 대한 연구와, 웹 상에서 이질적 어플리케이션들 간의 통신에 있어 요구되는 SOAP과 SCOAP 프로토콜의 적용 관련 연구가 지속되어야 한다.

참고문헌

- [1] Alin Deutsch, M.F. Hernandez, D.Suciu, "Storing Semi-structured Data with STORED", Proceedings of ACM SIGMOD Conference, Philadelphia, Pennsylvania, 1999.
- [2] J.Shanmugasundaram, K.Tufte, C.Zhang, G.He, D.J.DeWitt, and J.F.Naughon, "Relational Database for Querying XML Documents : Limitations and Opportunities", Proceedings of 25th Int'l Conference of VLDB, Edinburgh, Scotland, UK, 1999.
- [3] Simple Object Access Protocol(SOAP)1.1, W3Consortium, "http://www.w3.org/TR/SOAP"
- [4] Microsoft Corporation, "Simple Object Access Protocol (SOAP)", SOAP White Paper, "http://www.microsoft.com/korea/msdn/workshop/xml/general/SOAP_White_Paper.asp", 1999.
- [5] Simple CORBA Object Access Protocol(SCOAP), OMG, "ftp://ftp.omg.org/pub/docs/orbos/00-09-03.pdf"
- [6] Rogue Wave Software, Inc., "Rogue Wave XML-CORBA Link : XML Access to CORBA-based Systems", Technical White Paper, 1999.
- [7] 이호섭, 홍훈선, "분산환경에서의 CORBA와 XML의 연동구조", 한국정보과학회 학술발표논문집(A), 제28권, 제1호, pp.424~426, 2001.
- [8] 조혜영, 손충범, 조기형, 유재수, "XML 문서 관리를 위한 데이터 모델 설계", 한국정보처리학회 지식및데이터공학연구회 학술발표논문집, 제8권, 제1호, pp.137~143, 2001