2001년 한국정보처리학회 추계 학술발표논문집 제8권 제2호

# XML 기반 상품 표현 모델의 구현 및 분석

김경래, 하상호
순천향대학교 정보기술공학부
e-mail:krkim@cse.sch.ac.kr, hsh@sch.ac.kr

# Implementation and Analysis of a XML Based Product Description Model

Kyoungrea Kim and Sangho Ha
Dept of Informatin Technology, Soonchunghyang University

**요 약**

인터넷 중심의 정보화 사회가 되면서 B2C간 또는 B2B간에 상품 정보의 교환이 활발해지고 있다. 본 논문에서는 상품정보 교환을 위한 한 표준으로 이미 제안된 바 있는 XML 기반 통합 상품 표현 모델을 참조하고 구현한다. 이 모델은 다양한 상품정보를 XML에 기반하여 효과적으로 통합하여 표현할 수 있다. 구현은 Java의 컴포넌트 기술인 Java Bean과 EJB를 사용하여 이루어진다. 참조 모델을 사용하면 모든 상품에 공통된 데이터와 본질적인 데이터로 구분하여 기술할 수 있으며, 따라서 상품의 공통된 정보를 통합하여 기술함으로써 데이터의 중복을 피할 수 있다. 논문에서는 참조 모델이 갖는 데이터 중복 제거 효과를 웹 상의 다양한 상품 정보를 대상으로 분석한다.

## 1. 서론

인터넷의 확산과 더불어 B2C의 수요가 크게 증가하면서, 각각의 기업들에서는 B2B의 중요성을 깨닫게 되었고, 이를 위한 표준 거래 환경을 제정하기 위한 노력을 하고 있다[1]. 이와 같은 노력에 의해 제안된 프레임워크들은 기업간의 상품에 대한 정보 교환을 위해 XML[2]에 기반하여 각각의 고유한 상품카탈로그를 제공하고 있다. 상품카탈로그는 각 기업에서 출시하는 상품에 대한 정보를 표현하기 위한 구조를 가지고 있고, 표현되는 각각의 모든 상품들은 공통적인 정보와 고유의 정보를 가지고 있다.

이러한 모든 상품 정보를 통합적으로 표현할 수 있는 XML기반 통합 상품 표현 모델[3]이 지난 연구에서 제안된 바 있다. 이 상품 표현 모델은 XML에 기반하여 상품 정보를 효과적으로 통합하여 기술할 수 있는 표준 상품 표현 모델로서, 전자상거래에서 취급되는 대부분의 상품을 기술할 수 있으며, 상품 정보를 속성별로, 단계적 세분화를 통해서 기술 가능하며, 상품 정보상에 중복될 수 있는 데이터를 피하는 것이 가능하다.

본 논문에서는 이 상품 표현 모델을 Java의 컴포넌트 기술인 Java Beans와 EJB(Enterprise Java Beans)[4]에 기반하여 구현하고, 구현된 시스템을 실제 웹상의 상품 정보들을 대상으로 테스트한다. 또한, 시스템의 적용을 통해서 이 모델이 상품 정보 기술시 갖는 데이터의 중복 제거 효과를 분석한다.

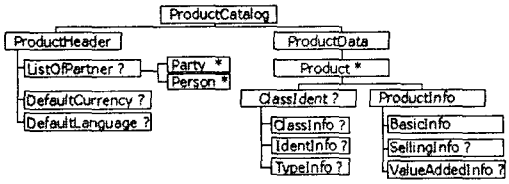
상품표현 모델의 구현시 사용되는 컴포넌트는 특정한 기능을 수행하기 위해 독립적으로 개발, 보급되고, 잘 정의된 인터페이스를 가지고, 다른 부품과 조립되어 응용시스템을 구축하기 위해 사용될 수 있는 소프트웨어의 단위로서, 소프트웨어의 재사용성과 다른 플랫폼과의 호환성, 이식성의 장점을 제공한다. 현재 컴포넌트 아키텍처 표준으로 Sun의 EJB와 MS의 COM(Component Object Model)[5]을 들 수 있는데, EJB가 COM보다는 기존 컴포넌트들을 더 많이 제공한다는 장점이 있고, 또한 현재 XML 기반의 애플리케이션 개발은 Java 기술을 사용하여 많이 개발되고 있기 때문에, 논문에서는 컴포넌트 기술로 Java의 EJB를 선택한다.

논문의 순서는 다음과 같다. 2장에서 XML기반 통합 상품 표현 모델에 대해서 간략히 서술하고, 3장에서는 이 모델의 구현 사항에 대해서 기술하고,

4장에서는 웹 쇼핑몰의 다양한 상품정보를 대상으로 모델이 갖는 데이터 중복 제거 효과를 분석한다. 마지막으로 5장에서 결론을 언급한다.

## 2. XML기반 통합 상품 표현 모델

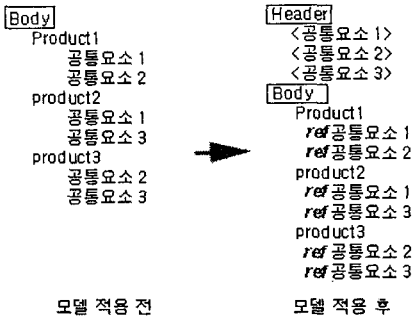
이 장에서는 XML 기반 통합 상품 표현 모델에 대해서 간략히 설명한다. 그림 1은 이 모델의 전체적인 구조를 보여준다.




<그림 1> XML 기반 통합 상품 표현 모델

상품정보는 크게 ProductHeader와 ProductData의 두 부분으로 구분된다. ProductHeader에는 기본적으로 모든 상품에 공통적으로 기술될 수 있는 정보가 포함된다. 여기에는 상품정보 기술에 사용되는 기본 언어, 통화, 상품의 제작이나 취급에 관련된 Partner 정보 등이 포함된다. 상품의 고유한 정보는 ProductData에 기술된다. 상품의 고유정보는 분류/식별정보(ClassIdent)와 상품정보(ProductInfo)로 구분되며, 상품정보는 기본정보(BasicInfo), 판매정보(SellingInfo), 부가가치 정보(ValueAddedInfo)로 세분화된다. 분류정보는 여러 가지 상품을 품목별로 분류하는 정보를 나타내고, 식별정보는 여러 가지 상품 중에 해당 상품을 식별하기 위한 정보를 나타낸다. 여기에 유형정보(TypeInfo)가 추가로 제공된다. 유형정보는 한 대표제품(예: 도서)내에서 각 개별제품들을 구분하고 유형화하는 기준을 제공하는 정보로, 주로 소비자의 탐색편의를 도모하기 위해서 제공된다. 상품의 기본정보에는 상품에 본질적인 기본적인 정보가 기술되고, 판매정보에는 가격이나 가격의 할인율 등과 같이 판매에 관련된 모든 정보가 기술되며, 부가가치정보에는 상품 평가 및 관련 상품 등과 같이 소비자의 제품 선택 편의와 합리적 선택을 도와주는 부가가치 정보가 기술된다.

앞서 설명하였듯이, ProductHeader에는 기본적으로 모든 상품에 공통적으로 기술될 수 있는 정보가 포함되는데, 이러한 사실을 이용하여 상품간의 공통된 정보를 ProductHeader 상에 기술함으로써 상품정보의 데이터 중복을 피할 수 있다. 그림 2는 이 모델이 데이터 중복을 어떻게 피할 수 있는지를 설명한다. Product1, 2, 3 상의 공통된 데이터를 Header 상에 위치시키고, 각 Product에서 그 데이터를 참조하도록 한다.



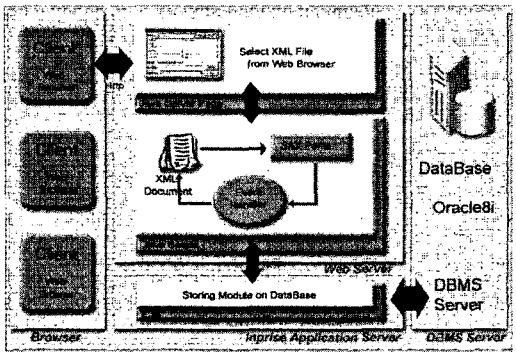

<그림 2> 상품정보의 중복데이터 제거

## 3. 모델의 구현

여기서는 2장에서 설명한 상품표현 모델의 구현 사항에 대해서 다룬다. 구현 환경으로, 시스템 서버에는 Sun OS 5.7의 운영체제가 사용되었고, 상품정보 저장을 위한 데이터베이스에는 Oracle8i[6]가 사용되었다. 또한, 웹 서버에는 Tomcat3.2.1 서버가, EJB를 위한 애플리케이션 서버에는 Boland사의 IAS4.0 서버가 각각 사용되었다. 프로그램 개발은 JBuilder4를 사용하여 이루어졌다.

시스템은 그림 3과 같이 크게 클라이언트, 미들웨어, 데이터베이스의 세 부분으로 구성된다. 클라이언트는 사용자와 시스템과의 인터페이스를 수행하는 웹 브라우저로 구성되며, 미들웨어는 클라이언트와 데이터베이스간에 가교 역할을 담당하는 시스템의 핵심 부분이며, 데이터베이스에는 실제의 데이터가 저장되고 관리되는 부분이다. 미들웨어에서 클라이언트와의 인터페이스는 JSP[7]로 구현되고, 웹브라우저와 EJB간의 인터페이스는 Java Beans로 구현되고, 데이터 베이스에 상품정보를 저장하는 모듈은 EJB로 구현된다. 이 저장 EJB는 Session Beans으로 구성되는데, Session Beans의 내부 메쏘드를 Java Beans에서 사용할 수 있도록, Remote Interface 클래스의 메쏘드를 사용한다.

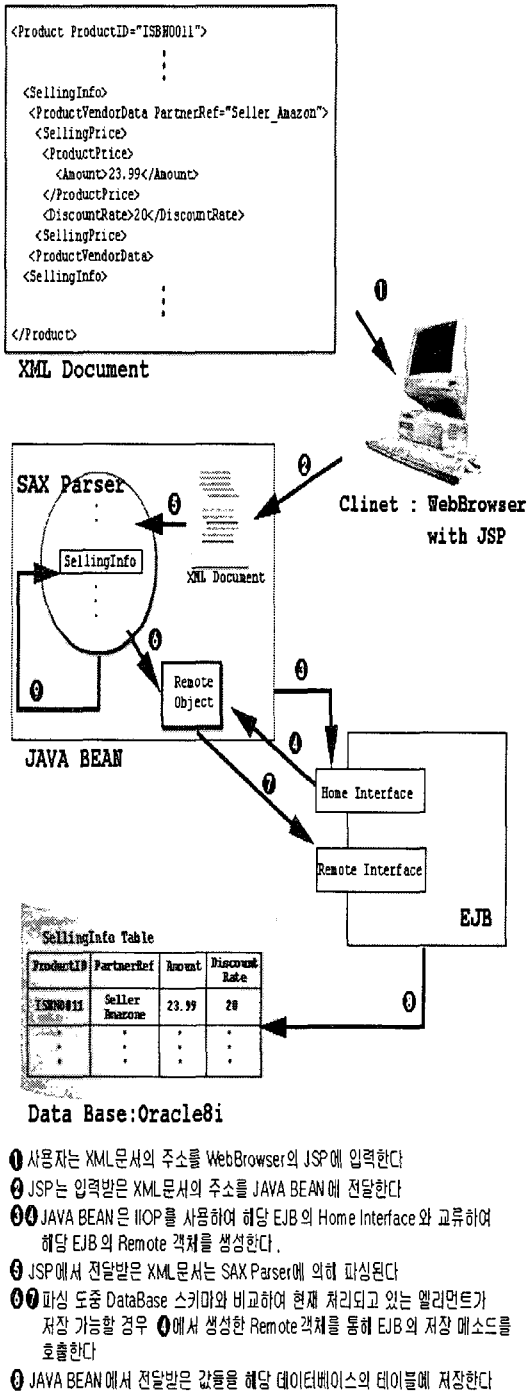
사용자는 웹 브라우저를 통하여 상품정보의 검색 및 저장을 요청할 수 있다. 사용자가 웹 브라우저를 통하여 Http로 JSP에게 요청을 전달하면, JSP는 Java Beans를 호출하고, Java Beans는 IIOP로 EJB의 Remote Interface와 교류하며 전달받은 사용자




<그림 3> 시스템의 전체 구성도

요청을 적절히 처리하고, 결과를 다시 JSP에게 다시 보낸다. JSP는 그 결과를 웹브라우저에 출력한다.

그림 4는 사용자로부터 2장에서 설명한 상품표현 모델에 따라 기술된 특정 상품정보에 대한 XML 문서를 시스템이 데이터베이스에 저장하는 과정을 보여준다. 먼저, XML 문서는 사용자의 웹 브라우저를 통하여 JSP 모듈에 의해 Java Bean을 호출한다. 이 때 XML 문서의 URL이 인수로 Java Bean에 전달된다. 호출된 Java Bean은 먼저 EJB와의 연결을 위해 리모트 객체를 생성한다. 이 객체가 생성된 후, 해당 XML 문서가 SAX 파서[8]에 전달되어 파싱이 시작된다. 파싱 과정에서 데이터베이스 스키마와의 비교가 이루어지며, 조건을 만족하는 XML 요소(element)를 만날 때, 앞서 생성한 리모트 객체를 통해 EJB의 저장 메쏘드를 호출하여 이 XML 요소의 내용을 데이터베이스의 해당 테이블에 적절히 저장될 수 있게 한다. 이 요소에 대한 처리가 끝나면, Parser는 다음 번째 XML 요소를 처리하기 시작한다. 이러한 과정은 XML 문서의 모든 요소를 처리할 때까지 반복된다. 그림 4에서는 XML 문서의 <SellingInfo> 요소가 SAX 파서에서 처리될 때, 이 요소가 데이터베이스의 SellingInfo Table로 어떻게 사상되는지를 보여준다. 파서는 SellingInfo Table의 스키마와 비교하여 저장될 정보들로 ProductID(ISBN0011), PartnerRef(Seller_Amazone), Amount(23.99), DiscountRate(20) 등을 추출하고, 이러한 데이터를 매개변수로 하여 EJB의 저장 메소드를 호출한다. 저장 메쏘드는 전달받은 데이터를 데이터베이스에 저장할 SQL 문으로 적절히 생성하고, JDBC를 통해서 데이터베이스 Oracle 8i와 연결한 후에, 생성한 SQL 문을 Oracle 8i에 전달하여 수행하도록 한다.




❶ 사용자는 XML문서의 주소를 WebBrowser의 JSP에 입력한다
❷ JSP는 입력받은 XML문서의 주소를 JAVA BEAN에 전달한다
❸❹ JAVA BEAN은 IIOP를 사용하여 해당 EJB의 Home Interface와 교류하여 해당 EJB의 Remote 객체를 생성한다.
❺ JSP에서 전달받은 XML문서는 SAX Parser에 의해 파싱된다
❻❼ 파싱 도중 DataBase 스키마와 비교하여 현재 처리되고 있는 엘리먼트가 저장 가능할 경우 ❹에서 생성한 Remote객체를 통해 EJB의 저장 메소드를 호출한다
❽ JAVA BEAN에서 전달받은 값들을 해당 데이터베이스의 테이블에 저장한다
❾ 다음 엘리먼트의 파싱을 수행한다

<그림 4> 시스템의 XML 문서 저장 과정 예

## 4. 분석

논문에서 참조하고 있는 상품표현 모델의 특징중의 하나가 상품간의 공통된 정보를 ProductHeader 상에 기술함으로써 상품정보의 데이터 중복을 피할 수 있음을 앞서 설명하였다. 여기서는 구현된 시스템을 사용하여 이러한 데이터 중복 제거 효과를 분석하고자 한다. 테스트 데이터는 웹 사이트 amazon.com에서 Computer 카테고리의 Java 키워드에 의해 검색된 도서들 중에서 40개를 순차적으로 선택하여 결정되었다.

중복 제거 효과 분석을 위해서, 2개의 상품 표현 모델을 생각한다: [3]에서 제안된 상품 표현 모델(모델 A)과 이 모델에서 공통부분을 따로 기술하지 않도록 약간 수정한 모델(모델 B). 각 모델에 대해서 상품정보 구조를 기술하는 DTD가 작성되었고, 위에서 선택한 40개의 도서 상품에 대해서 2개의 DTD에 따른 XML 문서가 각각 작성되었다. 즉, 모델 A에 따른 XML 문서와 모델 B에 따른 XML 문서가 생성되었다. 이러한 두 유형의 XML 문서를 구현한 시스템을 통해 데이터베이스 저장함으로써 두 모델을 비교한다. 비교 기준은 생성된 데이터베이스 테이블의 개수, XML 문서 저장에 걸리는 시간, 데이터 베이스의 필요한 기억공간, XML 파일의 용량이다. 표 1은 이러한 관점에서 두 모델을 비교한다.

<표 1> XML 문서 중복 데이터 제거 효과

| 모델 | 테이블 수 | XML 파일용량 | DB용량 | DB저장시간 |
|---|---|---|---|---|
| A | 23개 | 331Kb | 1064Kb | 31.3sec |
| B | 23개 | 468Kb | 1702Kb | 35.4sec |
| A/B | 0% | 30% | 37% | 12% |

중복 데이터를 제거한 모델 A가 그렇지 않은 모델 B보다 XML 파일의 크기가 30% 정도 작으며, 그 결과로 이 문서의 저장시 차지하는 DB 기억공간 용량이 37% 정도 절약할 수 있고, 데이터베이스에 저장하는 시간도 12%정도 단축할 수 있음을 보여준다. 그러나 생성된 테이블의 개수는 동일함을 보여준다.

중복 데이터의 제거 효과는 주로 상품의 제작이나 취급에 관련된 Partner 정보에서 온다. 도서의 경우에, 출판사와 저자들의 정보가 Partner 요소 상에 기술된다. 따라서 동일한 출판사가 여러 권의 도서를 출판하거나 동일 저자가 여러 권의 도서를 펴낸 정도에 따라서 그 중복 제거 효과는 달라진다. 논문에서 두 모델의 비교에 사용된 40개 도서가 17개의 출판사에서 제작되었다. 즉, 모델 B의 경우 한 출판사의 정보가 2.4번 중복되어 기술된다. 표 1의 데이터는 이러한 중복을 제거하는 결과로부터 해석된다.

테스트 데이터로 사용된 웹 상의 도서 정보는 참조 모델의 Partner에 관한 정보를 충분히 기술하지 않는다. 이러한 정보가 충분히 기술될 경우에 중복 제거 효과는 더 커질 것으로 판단된다. 또한, 40권의 도서 가운데서 동일 저자가 불행하게도 발견되지 않았다. 그러나 CD 음반의 경우 제작자, 기획사, 가수, 작곡자, 지휘자 등의 정보가 포함되는데 이러한 정보는 중복 가능성이 크며, 중복 제거 효과도 더 커질 것으로 본다. 결과적으로, 참조 모델이 갖는 중복 제거 효과는 상품의 특성에 크게 종속된다.

## 5. 결론

본 논문에서는 XML 기반 통합 상품 표현 모델을 참조하여 Java의 컴포넌트 기술인 Java Beans와 EJB를 사용하여 구현하였으며, 웹 상의 도서 상품에 대해서 그 모델이 갖는 상품정보의 중복 제거 효과를 분석하였다. 상품정보의 중복을 피함으로써, 상품정보의 효과적 작성, 데이터베이스의 공간 절약, 상품정보 저장 시간 단축, 네트워크 통한 파일 전송시간 단축 등의 효과를 가질 수 있다.

## 참고문헌

[1] 조현성 외 7인, "XML 기반 전자상거래 프레임워크 기술", 정보과학회지, 19권, 1호, p38, 2001. 1.

[2] "Extensible Markup Language (XML)", http://www.w3.org/XML/, 2000.

[3] 김경래, 하상호, 서건수, "XML 기반 통합 상품 표현 모델", 정보과학회학술대회, 2001. 4.

[4] Paul Tremblett, Instant Enterprise JavaBeans, Reading, McGraw-Hill,2001.1.

[5] http://www.microsoft.com/com/, COM: Delivering on the Promises of Component Technology

[6] Kevin Loney, George Koch, Oracle8i:The Complete Reference, Reading, McGraw-Hill, 2000.5.

[7] Duane K. Fields, Mark A. Kolb, Web Development with Java Server Page, Reading, MANNING, 2000. 5.

[8] Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto, XML and Java: Developing Web Applications, Reading , Addison-Wesley, 1999. 5.