

# 객체 모델 기반 XML 문서 변환을 위한 EJB 컴포넌트 설계 및 구현

김용수<sup>†</sup> · 임종선<sup>\*\*</sup> · 주경수<sup>\*\*\*</sup>

## 요 약

최근 소프트웨어 개발비용의 절감과 신뢰성 있는 소프트웨어의 개발을 위하여 컴포넌트 기반의 소프트웨어 개발에 대한 연구가 활발하게 진행되고 있다. 컴포넌트를 기반으로 한 소프트웨어 개발의 핵심은 컴포넌트의 품질이며, 컴포넌트의 품질을 결정짓는 가장 중요한 특성 중의 하나가 컴포넌트의 맞춤 용이성이다.

본 논문에서는 관계형 데이터베이스 검색 결과를 XML 문서로 변환하기 위한 EJB 컴포넌트를 설계 및 구현하였다. 이에 따라 사용자들은 관계형 데이터베이스 기반의 XML 응용 시스템을 개발하고자 할 때, 검색 컴포넌트를 단순·조립함으로써 시간단축 및 비용절감을 기대할 수 있다.

## Designing and Implementing EJB Component for Transform XML Documents based on Object Model

Kim Yong-Soo<sup>†</sup>, Lim Jong-Seon<sup>\*\*</sup> and Joo Kyung-Soo<sup>\*\*\*</sup>

## ABSTRACT

Nowadays for reliable software and cost reduction there are many research works on software development based on component. One of the challenge in designing a component-based system is determining which components are required and where they fit in the overall system architecture.

In this paper, we developed a EJB component for transforming RDB instances to XML documents. Accordingly users can build XML applications based on relational database just by assembling this component. Consequently they can reduce time and cost for developing their XML applications.

**Key words:** XML, EJB, component, RDBMS

## 1. 서 론

XML은 웹에서 구조화된 정보나 반-구조화된 정보를 교환하기 위해 표준 마크업 언어로 채택되어 가고 있다. 이러한 XML 문서의 광범위한 확산은 필연적으로 XML 정보의 효과적인 저장 관리 및 검색 문제가 발생하게 되었다. 따라서 현재 데이터베이스 학계 및 업계에서는 XML 정보를 데이터베이스에서 어떻게 원활하게 관리할 수 있는가에 대해 활발한

연구와 개발 작업이 진행 중에 있으며, 이러한 결과로 다양한 표준들이 개발되고 있다.

본 논문에서는 UML(Unified Modeling Language) 모델링 기법을 사용하는 소프트웨어 개발방법론인 CBD(Component Based Development) 방법론에 따라 컴포넌트를 설계하였다. CBD 기술은 소프트웨어 위기 극복으로부터 시작되었으며 소프트웨어 공학 기법으로써, 보다 높은 생산성과 고품질의 소프트웨어 개발을 목표로 한다. 컴포넌트는 재사용성과 상호 운용성이 뛰어난 소프트웨어 부품으로서 어플리케이션 개발자는 마치 하드웨어를 조립하듯이 원하는 기능이나 성능을 가진 컴포넌트를 선택하여 컴포넌트 시스템을 구성할 수 있게 된다. 이렇게

본 연구는 정보통신부의 ITRC 사업에 의해 수행된 것임

<sup>†</sup> 정회원, 순천향대학교 일반대학원 전산학과 석사과정

<sup>\*\*</sup> 준회원, 순천향대학교 일반대학원 전산학과 박사과정

<sup>\*\*\*</sup> 순천향대학교 정보기술공학부 교수

개발된 컴포넌트 기반 시스템은 일부의 부품을 개선된 부품으로 대체함으로써 품질이나 성능을 개선할 수도 있고 새로운 운용환경에 쉽게 이식할 수도 있다. 따라서 기술 변동에 능동적으로 대응할 수 있으며 타 시스템과의 상호 운용성도 보장된다. 이러한 CBD 기반 기술의 장점을 이용하여 관계형 데이터베이스에서 데이터를 검색하여 XML 문서로 변환할 수 있는 컴포넌트를 설계하고 EJB(Enterprise Java Bean)로 구현하였다. 따라서 본 논문에서 구현한 컴포넌트를 이용하여 플랫폼-독립적이며 아올러 DBMS-독립적인 XML 응용 시스템을 조립방식으로 구현할 수 있다.

본 논문의 제2절에서는 관련 연구를 다루며, 3절에서는 XML 문서 변환을 위한 EJB 컴포넌트 구조를 설명한다. 4절에서는 CBD 방법에 따른 설계를 논하고, 5절에서는 구현을 기술하며, 마지막으로 결론을 기술한다.

## 2. 관련 연구

XML은 1998년 2월에 인터넷상에서 구조화된 문서를 표현하기 위한 W3C에 의해 발표되었다[1]. 또한 XML은 SGML에서 파생된 것으로써 SGML의 사용하기 어렵고 복잡한 기능을 축소하고 인터넷에서 사용하기 용이하도록 만든 언어이며, HTML과는 달리 사용자가 임의의 태그를 정의하여 사용할 수 있으며, 문서의 구조 정보를 표현할 수 있는 특징을 가진다. XML의 이러한 특징 때문에 다양한 분야에서 XML을 이용한 연구가 이루어지고 여러 시스템들이 개발되고 있다.

많은 논문들이 복잡한 XML 문서를 관계형 데이터베이스에 저장하는데 초점을 맞추어 데이터베이스와 XML 문서의 연결에 대하여 발표되었다. 발표된 논문들의 내용을 분류하면 주어진 XML 문서를 관계형 데이터베이스에 어떻게 저장할 것인가 하는 제안들을 하고 있으며, 다른 한편으로는 주어진 XML 문서나 DTD로부터 정보의 의미에 대하여 데이터베이스 제약사항들을 다루는 방법을 제안하였다[2-4,8,13].

Oracle사와 Microsoft사와 같은 큰 벤더들도 XML 문서 관리를 위한 지원을 하고 있다. Oracle에서는 XDK(XML Developer's Kit), XSU(XML SQL Utility)를 이용하여 XML 문서에 대한 검색,

삽입, 수정, 삭제 기능을 지원하고 있으며[5], Microsoft에서도 SQLXML3.0[6]를 이용해 SQL Sever 2000에서부터 XML 문서에 대한 처리를 지원해 주고 있다. 각각의 XML 지원 정책은 자사 제품에 한정되어 있다.

본 논문에서 구현한 컴포넌트를 이용할 경우 Oracle, SQL Sever 뿐만 아니라 Sybase, PostgreSQL에 저장된 데이터 검색이 가능하다. 따라서 본 논문에서 구현한 컴포넌트를 이용하여 플랫폼-독립적이며 아올러 DBMS-독립적인 XML 응용 시스템을 조립방식으로 구현할 수 있다.

## 3. XML 문서변환을 위한 EJB 컴포넌트 구조

관계형 데이터베이스에 저장된 데이터의 검색은 3가지 조건을 이용하여 검색할 수 있다. 각각은 Key 값에 의한 검색, SQL 문에 의한 검색, 데이터베이스 테이블 전체 검색으로 나누어진다. 3가지 방법을 이용하여 검색된 데이터는 XML 문서 형태로 변환된다. 변환에는 검색된 결과를 XML 문서로 변형하는 작업을 DBMS2DOM에서 수행하며 변형에 필요한 정보는 Map 파일로부터 Map 객체를 만드는 MapFactory에서 얻어온다. 검색 구조는 그림 1과 같으며 각각은 자바 클래스로 이루어져 있고 EJB 컨테이너 안에서 동작한다.

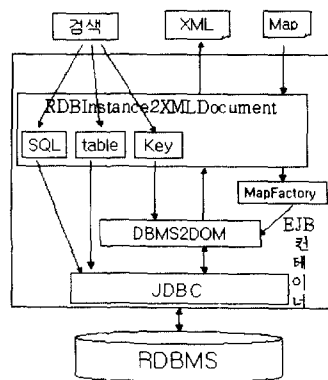


그림 1. XML 문서변환을 위한 EJB 컴포넌트 구조

### 3.1 XML 문서 검색

#### 3.1.1 Key 값에 의한 검색

그림 2의 Map 파일에서 "<ToRootTable>"으로

```
<ToRootTable>
  <Table Name="book"/>
  <CandidateKey Generate="Yes" <Column
Name="title"/> </CandidateKey>
  <OrderColumn Name="booksOrder" Generate="Yes"/>
</ToRootTable>
```

그림 2. book.map 파일의 일부

선언된 부분에서 테이블 이름을 book으로, 컬럼을 title로 선언하였다. 이것은 Key 값 검색의 테이블과 컬럼을 의미한다. DBMS2DOM에서 검색을 위한 SQL 문이 만들어지고 검색된 결과를 받아 Map 객체의 룰에 따라 XML 문서를 만든다.

3.1.2 SQL 문에 의한 검색

SQL 문에 의한 검색은 『select \* from book where title like 'java'』와 같은 형태로 book 테이블의 title에서 java를 포함하는 데이터를 검색한다. 검색된 데이터를 DBMS2DOM에서 Map 객체의 룰에 따라 XML 문서를 만든다.

3.1.3 테이블 검색

테이블 검색은 테이블 이름이 전달되며, book이라는 테이블 이름이 전달 될 경우 『select \* from book』형태의 SQL 문장을 만들어 테이블 전체의 내용을 가져오는 형태가 된다. 결국 테이블의 모든 데이터를 XML 문서로 변환하게 된다.

3.2 XML 문서 변환

3.2.1 MapFactory

MapFactory는 Map 객체를 만드는 곳으로 Map 안에 기술된 레지스트리 정보와 스키마 정보, 매핑 룰 등의 정보를 관리한다. 그리고 매핑 룰에 대한 매핑 테이블을 생성하고, DBMS2DOM이 검색 값에 대한 SQL 문을 생성하는데 필요한 정보 및 검색된 데이터를 XML 문서로 변환하는 정보를 제공한다.

그림 3의 Map 문서는 시스템이 자동으로 생성하기도 하며, 사용자 인터페이스를 통해서 나타나는 XML 문서의 구조를 표현한 DTD 트리와 테이블의 메타데이터 정보를 표현한 메타데이터 트리를 사용자가 보고 판단하여 관계 있는 것끼리 매핑을 시켜주는 방법으로 생성하게 된다[9,10,14].

```
<ClassMap>
<ElementType Name="book"/>
<ToClassTable><Table Name="book"/> </ToClassTable>
<PropertyMap>
  <Attribute Name="id"/>
  <ToColumn> <Column Name="id"/> </ToColumn>
</PropertyMap>
<PropertyMap>
  <ElementType Name="publication"/>
  <ToColumn><Column Name="publication"/> </ToColumn>
  <OrderColumn Name="publicationOrder" Generate="Yes"/>
</PropertyMap>
<PropertyMap>
  <ElementType Name="ISBN"/>
  <ToColumn> <Column Name="ISBN"/> </ToColumn>
  <OrderColumn Name="ISBNOrder" Generate="Yes"/>
</PropertyMap>
<PropertyMap>
  <ElementType Name="title"/>
  <ToColumn> <Column Name="title"/> </ToColumn>
  <OrderColumn Name="titleOrder" Generate="Yes"/>
</PropertyMap>
<RelatedClass KeyInParentTable="Candidate">
<ElementType Name="authors"/>
<CandidateKey Generate="Yes"><Column Name="bookPK"/>
</CandidateKey>
<ForeignKey><Column Name="bookFK"/></ForeignKey>
<OrderColumn Name="authorsOrder" Generate="Yes"/>
</RelatedClass>
</ClassMap>
```

그림 3. books.map 파일의 일부

3.2.2 DBMS2DOM

검색하여 얻어진 데이터는 MapFactory로부터 각 노드와 테이블간의 매핑 정보를 제공받아 XML 문서를 만든다.

Key 값에 의한 SQL문 생성

검색 값으로 title에 있는 JAVA를 검색 할 경우 DBMS2DOM은 모든 테이블을 조인하여 데이터베이스에서 데이터를 검색한다. 그림 4는 DBMS2DOM에 의해 생성되는 SQL 문장이다.

XML 문서 생성과정

① 검색한 데이터와 Map 객체의 매핑이 이루어진다.

```
select book.id, book.ISBN, book.title, book.publication,
       book.bookPK, authors.bookFK, authors.authorPK,
       author.author, author.authorFK
from book, authors, author
where book.bookPk = authors.bookFK and authors.authorPK
= author.authorFK and title = 'JAVA'
```

그림 4. Key 검색에 대한 SQL 문장

② Map 객체를 통해 book 이라는 ElementType이 book 테이블과 매핑 되고, id라는 컬럼은 Attribute id, 그리고 나머지 컬럼들이 ElementType으로 된다는 것을 알 수 있다.

③ <RelatedClass KeyInParentTable="Candidate">에 의해 bookPK는 자동 생성된 숫자를 가지고 authors 테이블과 연결이 된다는 것을 알 수 있다. 그러므로 authors는 DOM에서 자식을 가지는 노드가 된다. 그림 5는 테이블 구조를 보여 주고 있으며 그림 6은 검색한 결과를 XML 문서로 변환된 것이다.

#### 4. 컴포넌트 설계

본 논문에서는 UML 모델링 기법을 사용하는 소프트웨어 개발방법론인 CBD 방법론에 따라 컴포넌트를 설계하였다. CBD 기술은 소프트웨어 위기 극복으로부터 시작되었으며 소프트웨어 공학 기법으로써, 보다 높은 생산성과 고품질의 소프트웨어 개발을 목표로 한다. 즉, 생산성을 향상시키기 위해 컴포넌트를 만들고 재 사용할 수 있도록 하여 마치 부품을 조립하듯이 소프트웨어를 쉽게 개발할 수 있게 해 준다.

분석과정에서 컴포넌트 아키텍처 설계와 인터페이스 상호작용에 대한 모델링을 통하여 인터페이스

| book 테이블 |                |       |             |        |
|----------|----------------|-------|-------------|--------|
| id       | ISBN           | title | publication | bookPK |
| 11314    | 89-7467-8113-6 | JAVA  | 삼각지         | 2      |

| authors 테이블 |          |
|-------------|----------|
| bookFK      | authorPK |
| 2           | 3        |

| author 테이블 |          |
|------------|----------|
| author     | authorPK |
| 민계철        | 3        |

그림 5. 저장된 테이블 형태

```
<book id="11314" >
  <ISBN> 89-7467-8113-6</ISBN>
  <title>JAVA를 이용한 3D 만들기</title>
  <authors>
    <author>민계철,</author>
  </authors>
  <publication> 삼각지</publication>
</book>
```

그림 6. 검색결과로 생성된 XML 문서

들을 도출한다. 컴포넌트 개발에 있어서 중요한 작업은 바로 적절한 인터페이스를 정의하는 것이다. 설계 과정에서 인터페이스를 정의하기 위하여 다이어그램을 작성한다[11]. 본 논문에서는 정확한 인터페이스 도출을 위해 CBD 방법론을 이용하였고 향후 저장 컴포넌트와 같은 다른 컴포넌트와 조립이 용이하다.

#### 4.1 요구사항

##### 4.1.1 비즈니스 프로세스

그림 7에서 비즈니스 프로세스는 비즈니스 모델링하기 위해 전체적인 흐름을 보여준다.



그림 7. 비즈니스 프로세스

##### 4.1.2 비즈니스 개념 모델

비즈니스 프로세스 기술에 명확히 정의해야 할 필요가 있는 업무 용어를 산출하여야 한다. 산출한 업무 용어들을 전체적으로 도식화하는데, 이것은 비즈니스 개념 모델이라고 한다[11]. 따라서 실제 업무를 지원하기 위해 시스템이 관리해야 하는 정보를 표현할 수 있다. 그림 8에서는 여러 개의 관계형 데이터베이스 테이블에서 하나의 XML 문서로 변환할 수 있다.



그림 8. 비즈니스 개념 모델 다이어그램

##### 4.1.3 유즈케이스

유즈케이스는 시스템의 동적인 면을 모델링하기 위한 것으로 비즈니스 프로세스 모델에서의 프로세스 단계별로 책임을 할당하는 초기 작업을 진행할 수 있다. 유즈케이스 모델은 시스템과 상호 작용하는 액터(actor)를 포함하며, 액터는 또 다른 시스템을 나타내기도 한다[11]. 또한 유즈케이스 모델 내의 액터는 하나 이상의 유즈케이스에 관여한다. 그림 9에서는 'XMLDocument2RDBInstance' 기능을 user가 관여하고 있다는 것을 보여준다.

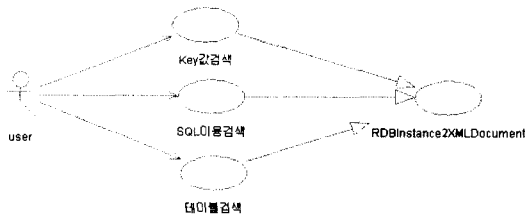


그림 9. 유즈케이스 다이어그램

4.1.4 유즈케이스 명세서

유즈케이스 다이어그램에서 유즈케이스, 액터, 그리고 그들 사이의 연관관계를 표시한다. 그 다음에 유즈케이스의 사건 흐름을 기술함으로써 유즈케이스 명세화하였다. 사건 흐름은 액터의 요청에 대해서 액터와 유즈케이스 사이에 어떤 상호작용이 발생하는지 보여준다.

4.2 컴포넌트 식별

컴포넌트 식별 단계는 요구사항 정의 워크플로우

|  |
|--|
| <p>이름 : Key값검색</p> <p>기동자 : user</p> <p>목표 : 관계형 데이터베이스에서 XML 문서를 생성<br/>주요시나리오</p> <ol style="list-style-type: none"> <li>1. 사용자는 검색 값과, Map 문서를 입력한다.</li> <li>2. 시스템은 입력받은 검색 값을 가지고 SQL 문장을 만든다.</li> <li>3. 시스템은 SQL 문에 따라 검색을 수행하여 값을 반환한다.</li> <li>4. 시스템은 입력받은 Map 문서의 규칙에 따라 검색된 값을 XML 문서로 표현한다.</li> </ol> <p>확장</p> <ol style="list-style-type: none"> <li>1.1 Map 문서가 아닐 경우 사용자에게 에러 메시지를 출력한다.</li> </ol> |
| <p>이름 : SQL이용검색</p> <p>기동자 : user</p> <p>목표 : 관계형 데이터베이스에서 XML 문서를 생성<br/>주요시나리오</p> <ol style="list-style-type: none"> <li>1. 사용자는 검색 조건 SQL 문, Map 문서를 입력한다.</li> <li>2. 시스템은 SQL 문에 따라 검색을 수행하여 값을 반환한다.</li> <li>3. 시스템은 입력받은 Map 문서의 규칙에 따라 검색된 값을 XML 문서로 표현한다.</li> </ol> <p>확장</p> <ol style="list-style-type: none"> <li>1.1 Map 문서가 아닐 경우 사용자에게 에러 메시지를 출력한다.</li> </ol>                                       |
| <p>이름 : 테이블검색</p> <p>기동자 : user</p> <p>목표 : 관계형 데이터베이스에서 XML 문서를 생성<br/>주요시나리오</p> <ol style="list-style-type: none"> <li>1. 사용자는 검색할 테이블을 입력한다.</li> <li>2. 시스템은 입력받은 Map 문서의 규칙에 따라 테이블의 모든 값을 XML 문서로 표현한다.</li> </ol> <p>확장</p> <ol style="list-style-type: none"> <li>1.1 Map 문서가 아닐 경우 사용자에게 에러 메시지를 출력한다.</li> </ol>  |

그림 10. RDBInstance2XMLDocument 유즈케이스 명세서

의 산출물인 비즈니스 개념 모델과 유즈케이스 모델을 입력으로 받아 작업을 진행한다. 컴포넌트 식별 단계의 목표는 초기 인터페이스와 컴포넌트 명세를 생성하여 컴포넌트 아키텍처의 초안을 만드는 것이다[11].

4.2.1 시스템 인터페이스 식별

각 유즈케이스마다 하나의 인터페이스와 하나의 다이얼로그 타입을 정의한다. 그 다음 각각의 유즈케이스 단계들을 하나씩 살펴보고, 각각의 단계들을 만족시키기 위해 어떤 기능을 제공해야 할 책임이 있는가를 판단하여 인터페이스를 찾는다. 그림 11에서는 시스템 인터페이스를 추출하는 과정을 보여준다. 그림 10에 있는 유즈케이스의 명세서는 각각의 단계에 따라 인터페이스를 추출할 수 있다.

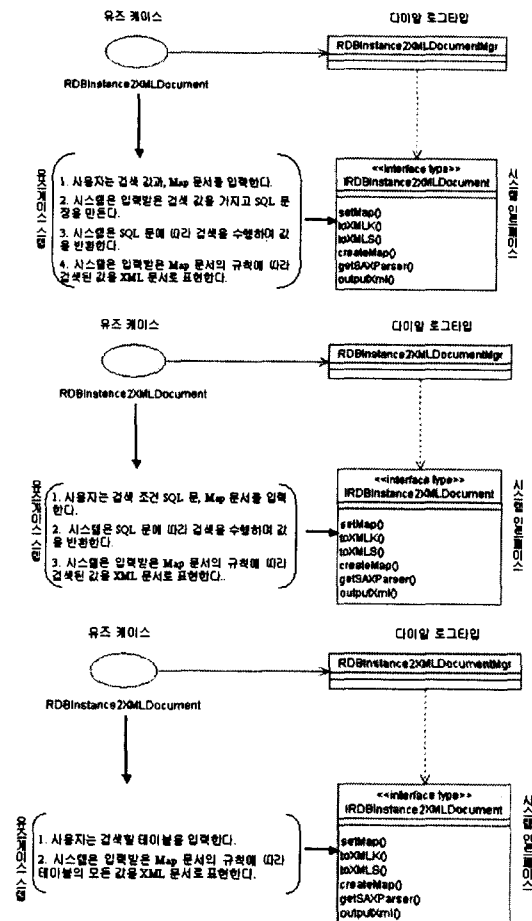


그림 11. 시스템 인터페이스와 유즈케이스의 매핑

4.2.2 비즈니스 인터페이스 식별

시스템에 의해서 관리되어야 하는 정보를 추상화 시켜 보여 준다. 비즈니스 인터페이스를 식별하기 위해서는 비즈니스 개념 모델에서 비즈니스 타입 모델로 생성하고 타입 중에 독립적으로 관리할 수 있는 모델을 핵심 비즈니스 타입으로 지정해 준다.

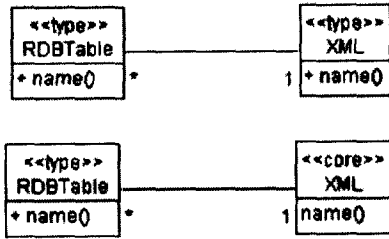


그림 12. 핵심 비즈니스 타입 모델 다이어그램

4.2.3 비즈니스 인터페이스 생성과 책임 할당

어느 정보가 어느 인터페이스에 의해서 관리되어야 할지를 명세 할 수 있다. 또한 인터페이스간의 의존성을 있는지를 명세화 하는데, 이를 위해서 상세 타입의 소유권을 적절한 인터페이스에 할당한다. 비즈니스 인터페이스 생성은 핵심 비즈니스 타입마다 하나의 인터페이스를 생성할 수 있다. 그림 13에서는 XML이 핵심 비즈니스 타입이므로, 하나의 인터페이스를 생성한다.

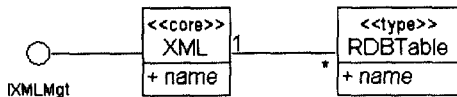


그림 13. 인터페이스 책임 다이어그램

4.3 컴포넌트 명세 아키텍처

초기 컴포넌트 명세를 생성하고 그것들이 어떻게 서로 어울리게 원리에 대한 것을 명세화 할 수 있다. 컴포넌트는 배포의 단위가 되며 컴포넌트 시스템의 교체의 단위이다. 또한 우리가 만들거나 구매해야 하는 공급의 단위, 즉 실체화(realization)의 단위이다. 따라서 사용자(애플리케이션 개발자)들을 선택적으로 컴포넌트를 교체하여 시스템을 개발할 수 있다 [11].

4.3.1 시스템 컴포넌트 명세

유즈케이스를 통해 도출된 시스템 인터페이스들은 중첩적이며 동일한 생명주기를 가지는 업무상의 사상들을 관리하고 있다. 이러한 인터페이스는 시스템 인터페이스들을 제공하는 하나의 컴포넌트 명세를 제안할 수 있다.

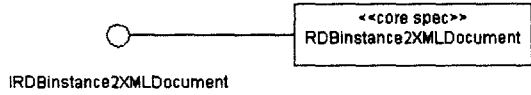


그림 14. 시스템 컴포넌트 다이어그램

4.3.2 비즈니스 컴포넌트 명세

식별된 비즈니스 인터페이스 당 한 개의 컴포넌트 명세를 작성할 수 있다. 인터페이스가 핵심 비즈니스 타입과 그와 연관된 상세 타입의 인스턴트들을 관리하기 위해 생성되었으므로, 이들 인터페이스들은 독립적으로 관리되어지는 정보와 관련이 있다.

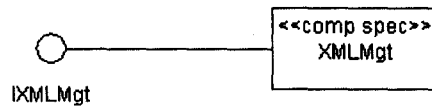


그림 15. 비즈니스 컴포넌트 다이어그램

4.3.3 컴포넌트 명세

그림 14에서는 상호 작용하는 컴포넌트간에 전달 되는 메시지들을 통해 그 연관성을 분석 할 수 있다. 따라서 컴포넌트 명세 아키텍처는 시스템의 정적인 부분을 표현하면서도 내부적으로 동적인 부분을 표현한다. 그림 16을 통해 시스템의 전체적인 구성을 살펴볼 수 있다.

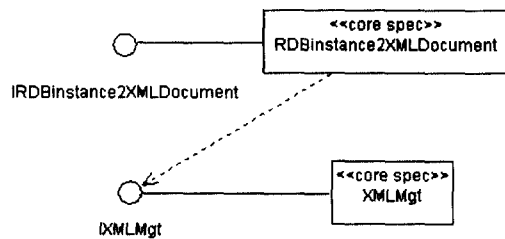


그림 16. 컴포넌트 명세 아키텍처

## 5. 구 현

EJB는 컴포넌트 기반 분산 객체 기술로서 엔터프라이즈급 애플리케이션 개발에 있어 추상 데이터와 비즈니스 로직에 대한 부분을 담당하는 매우 중요한 핵심 기술을 가지고 있다. 또한 개발자를 도와주는 EJB 컨테이너는 자동으로 엔터프라이즈 빈의 생명주기 관리, 상태 정보관리, 보안, 트랜잭션 처리, 영속성 처리 등을 포함한 수많은 내재된 서비스를 제공해주기 때문에 개발자는 비즈니스 로직만 담당하면 된다[12].

본 논문에서 구현한 컴포넌트명은 'RDBInstance2XMLDocument'라고 명명하였다. 본 컴포넌트는 관계형 데이터베이스의 검색 결과를 XML 문서로 변환하기 위해, EJB1.0 기반으로 하여 세빈 빈을 구현하였다.

### 5.1 컴포넌트 클래스 다이어그램

관계형 데이터베이스에서 검색을 위한 컴포넌트 클래스 다이어그램은 그림 19와 같다. 클라이언트 측에서 검색 쿼리와, 검색 조건에 해당하는 Map 문서를 EJB 컴포넌트에 보낸다. 컴포넌트는 클라이언트 쪽에서 넘겨준 값을 String으로 받게 되고 3절에서 설명한 방법을 이용하여 데이터베이스 테이블에서 검색을 하고 Map 객체의 규칙에 따라 XML 문서를 생성한다.

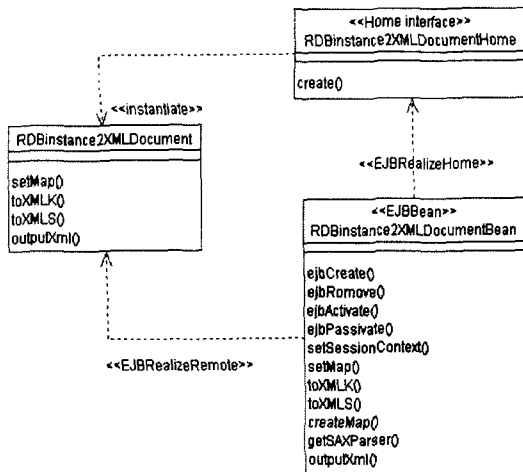


그림 19. RDBInstance2XMLDocument 컴포넌트 클래스 다이어그램

## 5.2 컴포넌트 인터페이스

### 5.2.1 Home 인터페이스

EJB 세션 빈에 접근하려는 모든 클라이언트는 홈 인터페이스를 통한다. 즉, JNDI 이름을 이용하여 세션 빈을 찾아낼 때 홈 인터페이스를 이용한다.

표 1. 홈 인터페이스

| 메소드      | 설명                            |
|----------|-------------------------------|
| create() | RDBInstance2XMLDocument 객체 생성 |

### 5.2.2 Remote 인터페이스

클라이언트가 세션 빈의 비즈니스 로직을 사용하기 위해서 리모트 인터페이스를 사용한다. 즉, 클라이언트가 홈 인터페이스의 create()를 호출하면 EJB 컨테이너는 리모트 인터페이스를 구현한 객체인 EJB 객체를 생성한다. 다음으로 EJB 객체의 비즈니스 메소드를 호출하면 동일한 이름을 가지는 메소드를 빈 인스턴스로부터 호출하여 서비스를 제공한다. 아래 표 2에서는 리모트 인터페이스와 각각의 동작을 설명한다.

표 2. 리모트 인터페이스

| 메소드  | 인자                                | 설명  |
|--|-----------------------------------|---|
| setMap(String)                                   | Map                               | 이 메소드는 컴포넌트 외부에서 읽은 Map 문서를 String 값으로 받아 임시 파일을 생성하는 메소드 |
| toXMLK(String, String, String, String, Object[]) | user, passwd, url, tableName, key | 데이터베이스 접속 정보, 테이블 네임과 검색 값을 인자로 받는다.                      |
| toXMLS(String, String, String)                   | user, passwd, url, sql            | 데이터베이스 접속 정보, 검색 SQL 문장이 전달된다.                            |
| outputXml()                                      |                                   | toXML에서 만들어진 문서 형태를 스트림 처리하여 만들어진 문서의 결과를 볼수 있게 한다.       |

### 5.2.3 EJB 빈 클래스

빈 클래스는 리모트 인터페이스에 선언된 메소드

가 실제로 구현되어 있는 클래스이다. 또한 내부적으로 쓰이는 자바 빈을 호출한다. 표 3은 리모트 인터페이스에서 선언한 메소드 이외의 빈 클래스의 메소드를 나타내고 있다.

표 3. 빈 클래스

| 메소드                                | 인자        | 설명  |
|------------------------------------|-----------|---|
| createMap(InputSource, Connection) | src, conn | toXMLK() 와 toXMLS()에서 호출되며, getSAXParser()을 호출한다. |
| getSAXParser()                     |           | createMap에서 호출되며, Map 문서를 파싱한다.                   |

### 5.2.4 세부 구현

'RDBInstance2XMLDocument' EJB 세션 빈은, 관계형 데이터베이스에서 검색된 값을 XML 문서로 변환하는 컴포넌트이다. 그림 2의'RDBInstance2XMLDocument' EJB 컴포넌트의 구조를 보여준다. 클라이언트에서 직접 EJB 서버에 파일 전송이 불가능하므로 Map 문서를 String으로 전송한다. 검색방법 따라 Key값, 테이블명, SQL 문장이 EJB 컴포넌트에 전달된다. 전달된 Map 문서와 검색 값의 처리 과정은 3절의 절차에 의해 이루어지며 사용된 핵심 자바 빈은 표 4와 같다. 본 컴포넌트에서 사용한 자바 빈은 XMLDBMS[7]에서 제공해 주는 자바 클래스들을 이용하여 구현하였다. XMLDBMS1.01은 XML 문서와 관계 데이터베이스 사이에 데이터를 전송하기 위한 자바 패키지이다.

표 4. 자바 빈 기능

| 자바 빈       | 기능   |
|------------|--|
| MapFactory | EJB 빈 클래스의 createMap()에서 호출되어 실질적인 Map 객체를 만드는 자바 빈이다. |
| DBMS2DOM   | 검색된 데이터를 Map 객체의 룰에 따라 XML 문서로 변환 하는 자바 빈이다.           |

### 5.3 테스트

본 논문에서 구현된'RDBInstance2XMLDocument' EJB 컴포넌트는 JBuilder4.0를 이용하여 컴포넌트를 구현하였고, 실행 환경으로서 EJB 컨테이너로 IAS

(Inprise Application Server)4.1에서'RDBInstance2XMLDocument'컴포넌트를 실행하였다. 'RDBInstance2XMLDocument'EJB 컴포넌트는 검색 조건에 따라 검색 질의가 컴포넌트에 전달되고 검색된 데이터는 XML 문서로 만들어져 파일로 생성된다. 그림 20은 테스트를 위해 Jbuilder의 실행창에 검색된 XML 문서가 출력되는 것을 볼 수 있다.

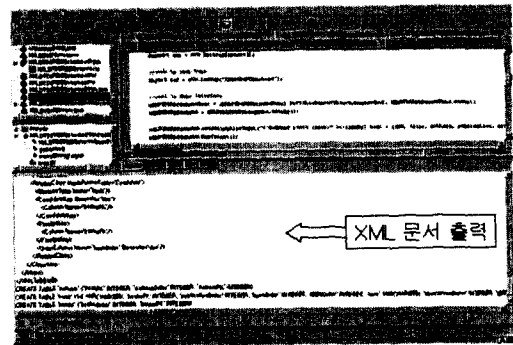


그림 20. 테스트 화면

## 6. 결론

XML은 웹에서 구조화된 정보나 반-구조화된 정보를 교환하기 위한 표준 마크업 언어로 채택되어 가고 있다. 이러한 현실 속에서 많은 양의 XML 데이터를 관리하기 위해서 관계형 데이터 베이스에 저장되고 있다. XML 응용 시스템에서는 검색한 데이터를 XML 문서로 변환할 필요가 있다. 본 논문에서는 플랫폼-독립적이며 아울러 DBMS-독립적인 XML 문서 변환을 위한 컴포넌트를 CBD 기반으로 설계하고, 이를 EJB 컴포넌트화 하는데 그 목적을 두고 있다. 이에 따라 본 XML 문서 변환 컴포넌트를 다른 컴포넌트들과 조합함으로써, 사용자들은 관계형 데이터베이스 기반의 XML 응용 시스템을 개발하고자 할 때 시간단축 및 비용절감을 기대할 수 있다.

## 참고 문헌

- [1] eXtensible Markup Language. <http://www.w3.org/TR/PR-XML-971208.html>.
- [2] D. Florescu and D. Kossmann, "Storing and Querying XML Data using an RDBMS." Data Engineering 22:3, pp.27-34, 1999.



[ 3 ] Kanne, C-C, and G. Moerkotte, "Efficient Storage of XML Data. In Sixteenth International Conference on Data Engineering(ICDE'00)." 28 February-3 March 2000, San Diego, IEEE Computer Society Press, Los Alamitos, CA, pp.198, 2000.

[ 4 ] M. Klettke and H. Meyer "XML and Object-Relational Database Systems-Enhancing Structural Mappings Based on Statistics." In Suciu, D., Vossen(eds.), G., Proceedings of the Third International Workshop on the Web and Databases, WebDB 2000, Dallas, Texas, USA, May 18-19, 2000, 63-68.

[ 5 ] Oracle Technology Network Korea, <http://otn.oracle.co.kr/tech/>

[ 6 ] SQLXML 3.0, <http://msdn.microsoft.com/downloads/default.asp?url=/downloads/sample.asp?url=/msdn-files/027/001/824/msdncompositedoc.xml>

[ 7 ] <http://www.rpbouret.com/xmldbms>

[ 8 ] 방승윤, 주경수. "XML 스키마를 관계형 스키마로의 변환에 관한 연구," 한국정보처리 지식 및 데이터 공학연구회 학술발표논문, pp.293-299, 2001.

[ 9 ] 이상태, 이정수, 주경수, "객체모델을 기반으로 한 XML DTD의 RDB 스키마로의 변환 방법", 대한전자공학회, 하계종합논문대회, 제24권 제1호, pp.113-116, 2001

[10] 이상태, 이정수, 주경수, "XML DTD를 기반으로 한, RDB 스키마 설계를 위한 Component 구현," 한국정보처리학회 지식 및 데이터공학연구회 제8회 학술발표대회 논문집, pp.309-316, 2001.

[11] 김경주, 조남규, UML Components 컴포넌트 기반 소프트웨어 명세를 위한 실용적인 프로세스, 인터비전, 서울, 2001.

[12] 유재우, 최재영, 최종명, 박준서, 프로그래머를

위한 EJB, 이한디지털리, 서울, 2001.

[13] 이상태, 주경수. "계약조건 유지를 위한 XML DTD의 관계 스키마로 변환 방법", 한국인터넷 정보학회, 제1권, 제2호, pp.189-196, 2000.

[14] 이정수, 방승윤, 주경수 "XML DTD의 관계형 데이터베이스 스키마로의 자동변환을 위한 컴포넌트 모델링", 인터넷정보학회논문지, 제2권 제5호, pp.81-90, 2001.



**김 용 수**

2001년 순천향대학교 전산학과 졸업(학사)  
 2001년~현재 순천향대학교 일반대학원 전산학과 재학 중(석사과정)  
 관심분야 : Database Systems, EJB, CBD

E-mail : admin@kimys.pe.kr



**임 종 선**

1997년 청운대학교 전산학과 졸업(학사)  
 1999년 순천향대학교 일반대학원 졸업(석사)  
 2002~현재 순천향대학교 일반대학원 전산학과 재학중(박사과정)

관심분야 : XML, XML Schema, UML.

E-mail : ronmer74@hotmail.com



**주 경 수**

1980년 고려대학교 이과대학 수학과 졸업(학사)  
 1985년 고려대학교 일반대학원 전산학과 졸업(석사)  
 1993년 고려대학교 일반대학원 전산학과 졸업(박사)  
 1986년~현재 순천향대학교 정보기술공학부 교수

관심분야: Database Systems, System Integration, Object-oriented Systems.

E-mail : gs00joo@asan.sch.ac.kr