

인트라넷에서의 웹 서비스를 이용한 이기종 RDB간의 데이터 이주

정희원 박 유 신*, 정 계 동**, 최 영 근***

The Migration of Data Between Heterogeneous RDBs Using Web Service in Intranet

Yoo Shin Park*, Kye Dong Jung**, Young Keun Choi*** *Reguler Member*

요 약

현재 기업 내의 정보 시스템들은 개별적이고 다양한 형태의 데이터베이스를 활용하여 업무적으로 발생하는 대량의 데이터를 저장하고 관리하고 있다. 이렇게 개별적으로 존재하는 데이터베이스에 저장된 데이터들을 이주시키기 위해 기업들은 EAI, MDR, DW 등의 기술을 활용하고 있다. 하지만 이러한 기술들은 도입 비용 및 유지관리에 많은 비용을 요구할 뿐만 아니라 각 벤더들마다 요구하는 환경이 상이하다는 문제점을 가지고 있다. 이러한 기존 기술들의 문제점을 해결하기 위해 본 논문에서는 웹 서비스를 기반으로 이기종 관계데이터베이스 간의 데이터 및 의미적인 제약 조건을 이주하는 데이터 이주 시스템을 설계한다. 웹 서비스를 사용함으로써 기업들은 기존의 웹 환경을 활용할 수 있으므로 도입 비용 및 유지관리 비용을 절감할 수 있고 각각의 시스템들은 XML 포맷을 사용함으로써 플랫폼 및 시스템 환경, 구현언어에 독립적으로 데이터를 이주할 수 있다.

Key Words : Data Interchange, Data Migration, Data Transformation, Web Service, XML

ABSTRACT

Information systems in current corporations are managing and storing large data officially happened on individual and various databases. Corporations for migrating the stored data from these individual databases are adopting with technologies of EAI, MDR, DW etc.. However these technologies are not only required to expenses introduction charge and maintenance cost but also have problems of heterogeneous environment required per each vender.

In this paper, to solve problems of these current existing technologies, we propose to design our data migration system to migrate source data and semantic constraint condition between heterogeneous relation databases based on web service. As corporations use web services, they can reduce introduction expense and maintenance cost because of using current existent web environment. Each system can independently migrate XML based data against any platform, system environment, and Implementation language.

I. 서 론

현재 대부분의 기업들은 다양한 종류의 데이터베이스를 활용하여 데이터를 저장 및 관리하고 있다.

이렇게 산재되어 있는 데이터베이스들의 데이터를 보다 효율적으로 관리하기 위해 기업들은 데이터 통합을 시도하고 있다. 데이터를 통합하기 위해 현재 적용되고 있는 방법은 두 가지가 있다. 하나는

* 근로복지공단 정보시스템실 (love0840@empal.com), ** 광운대학교 (gdchung@kw.ac.kr), *** 광운대학교 (ygchoi@cs.kw.ac.kr)
논문번호 : KICS2005-05-203, 접수일자 : 2005년 5월 20일

기업 내의 모든 데이터를 하나의 물리적인 데이터베이스 공간에 저장하는 방법과 다른 하나는 데이터베이스들 간에 연결을 형성하여 가상적으로 단일 데이터베이스처럼 보이도록 하는 방법이다. 하지만 전자의 경우는 특정 데이터베이스 내에 존재하는 모든 데이터를 통합 데이터베이스로 이관하여야 하며 그 이관대상으로 데이터뿐만 아니라 테이블, 인덱스, 제약조건 등 데이터베이스의 객체에 대한 이관도 수행하여야 하기 때문에 작업자는 많은 시간과 노력이 요구되는 과도한 수작업을 수행하여야 한다. 후자의 경우는 전자의 경우와 같은 과도한 수작업은 요구되지 않으나 각 데이터베이스간의 연결을 형성하고 데이터를 주고받기 위해 EAI, MDR 등의 기술을 활용하는 솔루션을 사용하고 있으나 이러한 솔루션들은 단일 벤더의 솔루션이란 한계점과 유지관리 비용의 증가라는 문제점을 가지고 있다. 이러한 상황에서 기업들은 기존의 방법들에 비해 작업량을 줄이면서 보다 유연하게 기업 내의 데이터를 통합하고 이관할 수 있는 방법을 요구하고 있으며 그와 같은 요구를 충족시키기 위한 방법으로 최근 웹 서비스가 주목받고 있다.

본 논문에서 설계하고 구현하는 데이터 이주 시스템은 사용자가 필요로 하는 데이터가 어느 관계 데이터베이스에 존재하는지 쉽게 검색하여 찾을 수 있고 표준화된 방법으로 데이터 및 데이터베이스 객체를 이관할 수 있도록 하는 시스템으로 테이블, 뷰, 인덱스, 키 정보 및 제약조건을 이주 대상으로 하며 각 RDBMS들의 데이터 타입 및 객체 명명법 등의 형태가 다르므로 이주 대상 객체의 정보 및 데이터를 이기종 관계 데이터베이스 간에 주고받기 위한 하나의 표준화된 매개체로 관계 데이터베이스의 의미적 제약조건 및 데이터를 표현할 수 있는 XML 이용하고 이주 대상 객체의 정보 및 실제 데이터를 매개체인 XML문서로 변환하는 컴포넌트와 XML문서를 파싱하여 타겟 데이터베이스에 이주된 객체를 생성하고 생성한 객체에 실제 데이터를 저장하는 컴포넌트를 EJB를 사용하여 구현한다.^[2]

본 논문에서의 사용자가 특정한 객체 및 데이터를 선택해 이주하여 활용할 수 있도록 하는 이주 컴포넌트 서비스를 제공한다. 즉 현재 운영되고 있는 DBMS의 객체 및 데이터들을 대상으로 신규 프로젝트에서 필요로 하는 데이터를 쉽게 검색하고 취득할 수 있는 서비스를 제공하는 것이다. 일반적인 데이터 이주의 장점으로는 이주 작업 완료 후 하나의 DBMS만을 관리하면 된다는 것이고 단점은

작업을 수행하는데 많은 시간과 노력이 요구되며 두 DBMS를 완전히 하나의 DBMS로 통합하기 위해 DB 설계까지 변경하는 경우가 있으며 JDBC나 ODBC와 같은 표준화된 DBMS 인터페이스를 이용한다 할지라도 각 DBMS에 의존적인 SQL문을 사용한 경우 SQL문을 수정해야 한다는 것이다.

본 논문에서 데이터 이주 작업을 위한 고려사항은 다음과 같이 범위를 제한 한다.

- 각 DBMS에서 지원하는 데이터 타입이 다르다. 각 DBMS들은 ANSI SQL 타입표준을 기반으로 하나 독자적인 데이터 타입을 사용하고 있다.
- 무결성 제약조건인 기본 키, 외래 키, 유일키, CHECK, DEFAULT 제약사항 등의 문장 형식은 유사하나 SQL Server에서는 군집색인이 디폴트로 생성되나 오라클에는 없다.
- NULL 값의 처리를 고려해야 한다. SQL Server는 유일키에 NULL 값의 중복을 허용하지 않으나 오라클의 경우에는 중복을 허용한다.
- 색인(INDEX)명칭의 부어 범위에 차이가 존재한다. SQL Server는 테이블 내에서만 유일하면 되나 오라클에서는 사용자 내에서 동일한 색인명을 허용하지 않는다.
- 보안 사항 처리 및 권한 관리 등을 고려해야 함.

위와 같이 데이터 이주 시에는 여러 가지 고려해야 할 사항들이 있다. 이러한 많은 고려사항들은 작업자에게 과도한 수작업을 요구하며 작업상의 오류를 유발할 수 있다.

표 1은 본 논문에서 이기종 RDB간의 데이터 이주 시 문제점과 해결 방안이다.

본 논문의 구성은 2장은 관련연구를 서술하고 3장에서는 웹 서비스를 이용한 데이터 이주 시스템의 설계를 기술하며 4장에서는 데이터 이주 시스템의 구현 및 테스트를 기술하고 5장에서는 결론을 기술한다.

II. 관련 연구

2.1 웹 서비스(Web Service)

XML을 기반으로 하는 웹 서비스는 표준화된 XML 메시지를 통해 네트워크 상에서 접근 가능한 메소드를 기술하는 인터페이스로 정의되며 분산 환경의 표준 프로그래밍 모델로 원격지의 클라이언트

표 1. 이기종 RDB간의 데이터 이주 시 문제점 및 해결방안

문 제 점	해 결 방 안
이기종 RDB 스키마간의 데이터 타입이 서로 상이하여 이관 작업이 어렵다.	이기종 RDB간의 타입 매칭 테이블을 변환 컴포넌트 안에 구성하고 중간 매개체인 XML 문서 생성시 반영하도록 한다.
이관 대상 객체의 속성 정보를 동일하게 생성하기 위한 수작업이 필요하다.	이기종 RDB간의 중간 매개체인 XML 스키마문서의 엘리먼트 및 어트리뷰트를 활용하여 객체의 속성 정보로 표현한다.
실제 데이터를 dump File 또는 Text File 형태로 변환하며 데이터를 구분하기위해 구분자를 이동한 후 수동으로 스키마를 생성 후 타겟 DB에 저장하기 위한 추가적인 작업이 요구된다.	XML 인스턴스에 실제 데이터를 저장하여 XML 스키마문서와 같이 SOAP를 통해 전송하여 DB 스키마와 실 데이터를 타겟 DB에 저장한다.
사용자가 필요로 하는 객체 및 데이터가 어느 RDB에 존재하는지 쉽게 알 수가 없다.	객체 정보 및 데이터를 제공하는 서비스를 UDDI에 등록하여 사용자가 UDDI 검색을 통해 쉽게 찾을 수 있도록 한다.

에 의해 호출될 수 있는 원격 프로시저의 집합 형태의 서버 응용 프로그램으로 웹상에서 플랫폼과 구현언어에 독립적인 컴포넌트 기반의 컴퓨팅 서비스이다.^[1]

기존의 분산 컴퓨팅에 사용되어 온 CORBA의 IIOP, DCOM, RMI 프로토콜은 연결지향 프로토콜로 요청이 들어오면 해당 원격 객체의 메소드를 실행시켜 결과를 전송하는 모델로 다음과 같은 문제점을 가지고 있다. 연결된 상태에서 갑작스런 연결 끊김, 서버 자원의 낭비, 방향변 통과의 어려움, 사용상의 어려움, 복잡한 구조 등의 문제점을 가지고 있다.^[3] 웹 서비스는 기존 기술들의 이러한 문제점을 해결하고 적은 비용으로 서로 다른 시스템들을 통합시키는 새로운 분산 컴포넌트 기반의 컴퓨팅 기술로 HTTP, XML, SOAP, WSDL, UDDI 등의 표준화된 기술을 이용하여 플랫폼 독립적이며, 시스템 상호운용성을 보장한다.^[4]

UDDI는 웹 서비스의 공개(Publish)와 검색(Find)을 위한 XML 레지스트리의 구현 및 기능과 사용 방법을 표준화한 것으로 제공자가 웹 서비스를 공개하는 방법 및 소비자가 웹 서비스를 검색하는 방법에 관한 메커니즘을 제공한다.^[5] WSDL(Web Service Description Language)는 표준화된 방식으로 웹 서비스의 인터페이스를 기술하는 XML 기술로 원격 프로시저를 이용하기 위해 매개변수, 프로시저명, 반환형을 기술하는 방법을 표준화한 것이다. WSDL 문서에서 생성된 코드는 웹 서비스 시스템에 바인딩 하여 SOAP 메시지를 생성하고, 전송시키는 역할을 수행한다.^[6]

SOAP(Simple Object Access Protocol)는 원격 응용프로그램 간의 정보를 교환하는데 사용되는 메시지 형식을 표준화한 것으로 XML 문서를 기반으로 하고 있으며 일반적으로 HTTP를 전송용 프로토

콜로 사용한다.^[7] 웹 서비스는 이들 기술을 기반으로 XML 데이터를 효율적으로 교환할 수 있고 적은 이식 비용 및 플랫폼 독립적이라는 장점을 갖는다.^[8]

2.2 XML과 RDB 변환기법

XML 문서의 계층적 구조는 자동화된 변환 규칙을 사용하여 관계 데이터베이스의 평면적 구조로 변환할 수 있다.^[15] XML 문서는 반 구조적인 문서로 관계 데이터베이스로 변환할 경우 많은 널(NULL) 값을 데이터로 갖는 테이블을 생성하게 된다. 이러한 문제를 해결하기 위해 문서를 하나의 테이블로 변환하는 대신 여러 개의 테이블로 변환하여 저장한다.^[9] 일반적으로 XML 문서를 관계 데이터베이스로 변환하기 위한 기법으로 테이블 기반 변환 기법과 객체-관계 변환 기법을 사용한다.^[10]

XPERANTO^[11]에서 제시한 테이블 기반 변환 기법은 XML 문서를 하나의 테이블 또는 테이블 집합으로 모델링하는 것으로 테이블을 나타내는 엘리먼트를 선언하고 해당 엘리먼트의 자식 엘리먼트로 컬럼을 선언하는 기법으로 테이블들을 XML 문서로 변환 시 테이블간의 관계를 표현하지 못하고 변환 규칙에 맞지 않는 XML 문서에 대해서는 사용할 수 없는 단점이 있다.

객체-관계 변환 기법은 XML 문서 내의 어트리뷰트, 엘리먼트, 컨텐츠, 혼합된 컨텐츠 등의 엘리먼트 타입은 클래스로 모델링되고 #PCDATA 또는 말단 엘리먼트 및 어트리뷰트 등은 스칼라 속성으로 모델링 된다. 이러한 모델링 후 클래스는 테이블로 변환되고 스칼라 속성은 테이블의 컬럼으로 변환된다. 객체-관계 변환 기법으로 대표적인 것은 XML-DBMS^[12], Inlining^[13], Extended Inlining^[14] 기법이 있다.

III. 웹 서비스를 이용한 이기종 RDB간의 데이터 이주 시스템 설계

본 장에서는 기업 내부의 인트라넷 상에서 웹 서비스를 이용하여 데이터를 이주하는 데이터 이주 시스템을 설계한다. 그림 1은 데이터 이주 시스템의 전체 구성도이다. 본 시스템은 세부적으로 구성된다. 첫 번째 부분으로 XML 문서와 관계 데이터베이스 간의 상호 변환을 위한 부분이다. 두 번째 부분은 각 로컬 시스템들이 공개하는 원격 프로시저의 정보를 저장하고 검색할 수 있도록 하는 UDDI 레지스트리 부분이다. 세 번째 부분으로 각 로컬 시스템들의 원격 프로시저를 호출하여 사용할 수 있도록 하는 웹 서비스 부분이다.

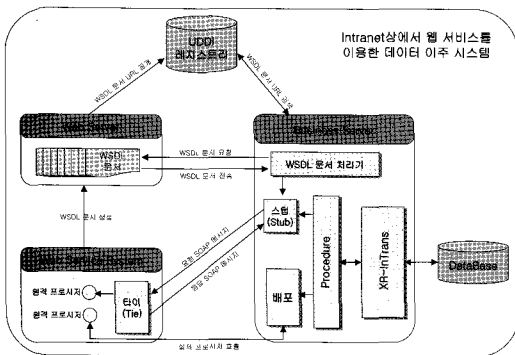


그림 1. 웹 서비스를 이용한 데이터 이주 시스템 전체 구성도

3.1 XML과 RDB 간의 변환 컴포넌트

XML과 RDB 상호간의 변환 작업을 수행하는 컴포넌트인 XR-InTrans는 RDBToXML 변환과 XMLToRDB 변환 부분으로 구성되며 그림 2는 전체 컴포넌트 구성도이다.

RDBToXML 변환은 관계 데이터베이스의 메타 데이터와 데이터를 XML Schema 문서와 XML 인스턴스 문서로 변환하는 컴포넌트로 세부 구성요소는 다음과 같다.

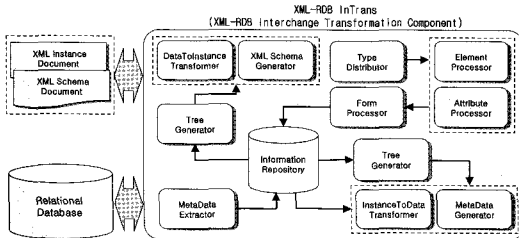


그림 2. XR-InTrans 컴포넌트 구성도

▪ **MetaData Extractor** : 테이블 또는 뷰의 컬럼 및 키, 인덱스 정보, Null 허용여부 등의 의미적 제약 조건을 포함하는 메타데이터를 추출하여 정보 저장소에 저장한다. 정보 저장소에는 각 메타데이터의 정보를 저장하기 위한 테이블 형태의 컬럼 정보, 인덱스 정보, 제약조건 정보가 존재하며 컬럼 정보는 테이블 명, 컬럼 명, 각 컬럼의 데이터 타입, 전체 길이, 소수점 자릿수, NULL 값 허용여부, 디폴트 값 등의 정보를 저장한다. 인덱스 정보는 테이블 명, 인덱스 명, 인덱스에 포함되는 컬럼 및 순서에 관한 정보가 저장되며 제약조건 정보는 기본 및 외래키의 선언 명, 키에 포함되는 컬럼 및 순서, 참조 테이블 및 참조 컬럼 정보 등이 저장된다. 표 2는 정보 저장소에 포함된 정보 테이블들의 구조이다.

표 2. 정보 저장소에 포함된 정보 테이블들의 구조

<테이블 컬럼 정보 구조>

항 목	설 명
테이블명	선택된 테이블의 명칭
컬 럼 명	테이블에 포함되어 있는 컬럼의 명칭
타 입	해당 컬럼의 데이터 타입
길 이	해당 컬럼의 전체 길이
순 서	해당 컬럼의 테이블상의 위치
소수점 자릿수	해당 컬럼의 소수점이하 자릿수
NULL 허용유무	해당 컬럼의 NULL 값 허용 유무
디폴트 값	해당 컬럼의 디폴트 값

<제약조건 정보 구조>

항 목	설 명
테이블명	테이블의 명칭
제약조건명	테이블에 정의된 제약 조건 명칭
타 입	정의된 제약 조건 구분('P','F')
컬 럼 명	제약조건에 포함된 컬럼의 명칭
순 서	제약조건에 포함된 컬럼의 순서
참조 테이블명	참조하는 테이블의 명칭
참조 컬 럼 명	참조하는 컬럼의 명칭
참조순서	참조하는 컬럼의 순서

<인덱스 정보 구조>

항 목	설 명
테이블명	테이블의 명칭
인덱스명	테이블에 정의된 인덱스 명칭
컬 럼 명	인덱스에 포함된 컬럼의 명칭
순 서	인덱스에 포함된 컬럼의 순서

표 3. 트리의 노드 구조

항 목	설 명
노드명	노드의 명칭
노드구분	노드 구분 값
컬럼타입	노드에 포함된 컬럼의 타입
컬럼크기	노드에 포함된 컬럼의 전체길이
소수점 자릿수	노드에 포함된 컬럼의 소수점 자릿수
NULL 허용유무	노드에 포함된 컬럼의 NULL 허용유무
디폴트 값	노드에 포함된 컬럼의 디폴트 값
원 테이블명	노드에 포함된 컬럼이 포함된 테이블명
기본키 여부	노드에 포함된 컬럼의 기본 키 여부

- Tree Generator : 정보 저장소의 정보를 이용하여 트리를 생성한다. 트리의 각 노드는 XML Schema 문서의 엘리먼트 또는 어트리뷰트로 변환 시 사용할 정보를 포함한다. 트리의 각 노드는 노드명과 변환 시에 사용될 노드 구분 정보와 데이터 타입 사상을 위한 부가 정보 및 컬럼이 포함된 테이블 명, 기본 키 유무 정보 등을 가진다. 표 3은 트리의 각 노드의 구조이다.
- XML Schema Generator : 생성된 트리과 정보 저장소에 존재하는 타입 매핑 정보를 이용하여 XML Schema를 생성한다. XML Schema 문서는 테이블 간의 관계 및 구조 정보의 의미적인 제약 조건 정보를 표현하기 위해 XML Schema의 아이덴티티 제약 조건 메커니즘인 Unique 메커니즘과 Key와 KeyRef 참조 메커니즘을 사용한다.
- DataToInstance Transformer : 테이블의 실제 데이터를 추출하여 XML 인스턴스 문서를 생성하기 위해 XML Schema Generator에서 생성된 트리 및 의미적 제약 조건 정보를 이용한다. 추출된 데이터는 트리의 계층 구조를 참조하여 데이터의 위치를 설정하고 제약조건 정보에 저장되어 있는 기본 키 정보를 통해 해당 컬럼이 엘리먼트로 작성될지 어트리뷰트로 작성될 것인지를 결정한다. XML 인스턴스 문서를 생성할 때 각 테이블의 기본 키 및 외래 키 정보나 인덱스 정보는 테이블의 활용도를 높이기 위해 설정된 정보이므로 실제 데이터를 교환하는데 사용되는 XML 인스턴스 문서에는 포함하지 않는다. 그림 3은 생성된 XML 인스턴스 문서의 예를 보이고 있다.

그림 4는 RDBToXML 변환의 각 구성요소들의 처리를 통해 XML Schema가 생성되는 과정이다.

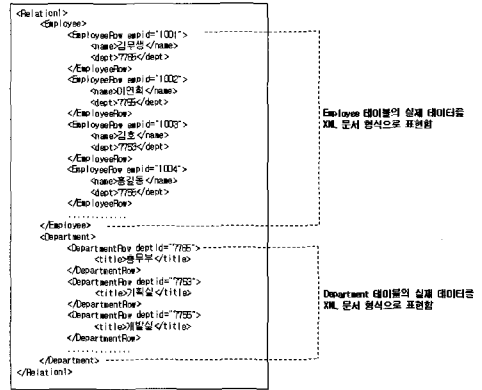


그림 3. 생성된 XML 인스턴스 문서의 예

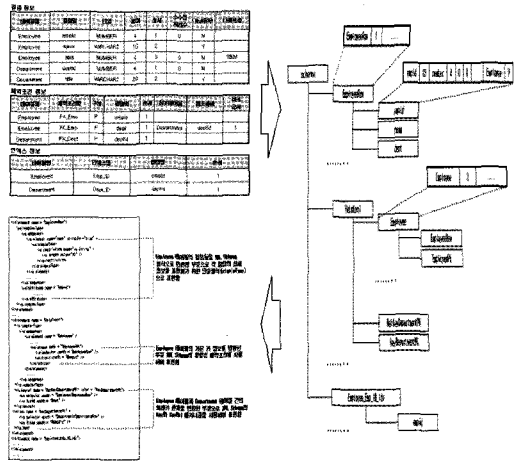


그림 4. RDBToXML 변환의 각 구성요소들의 처리 과정의 예

관계 데이터베이스와 XML Schema에서 제공하는 데이터 타입 간에는 차이가 존재하며 관계 데이터베이스의 키 및 인덱스 정보, Null값 허용 여부 등의 의미적인 제약 조건 정보를 XML Schema로 변환하기 위해서는 매핑 과정이 필요하다.

표 4는 데이터 타입 및 의미적 제약 조건 간의 사상을 나타내는 내용의 일부분이며 XML Schema의 데이터 타입 표현 시 보다 상세하고 세부적인 표현을 위해 단순형식(simpleType)을 사용한다. 이러한 단순형식을 사용함으로써 보다 정확한 의미적인 제약 조건과 메타데이터 형식 정보를 교환할 수 있다.

XMLToRDB 변환은 XML Schema와 XML 인스턴스를 의미적 제약 조건이 포함된 메타데이터와 데이터로 변환하여 관계 데이터베이스에 저장하는 컴포넌트로 첫째, XML Schema 문서를 분석하여 얻은 정보들을 정보 저장소에 존재하는 각 참조 정보에 저장하는 부분이다. 둘째, 정보 저장소에 저장

표 4. 관계 DB와 XML Schema 사이의 데이터 타입 사상 예

Oracle	SQL Server	XML Schema
Char(n)	Char(n)	<xs:restriction base="xs:String"> <xs:length value="n"/> <xs:pattern value="\d{n }"/> </xs:restriction>
VarChar2(n)	VarChar(n)	<xs:restriction base="xs:String"> <xs:length value="n"/> <xs:minLength value="0"/> <xs:maxLength value="n"/> </xs:restriction>
Number	float	<xs:element type = "xs:double"/>
Number(n)	decimal(n), numeric(n)	<xs:restriction base="xs:decimal"> <xs:totalDigits value="n"/> <xs:fractionDigits value="0"/> </xs:restriction>
Number(n,m)	decimal(n,m), numeric(n,m)	<xs:restriction base="xs:decimal"> <xs:totalDigits value="n"/> <xs:fractionDigits value="m"/> </xs:restriction>
Date	datetime	<xs:element type = "xs:datetime"/>
Primary Key	Primary Key	<xs:unique name = "EmployeePK"> <xs:selector xpath = "EmployeeRow"/> <xs:field xpath = "@empid"/> </xs:unique>
Foreign Key	Foreign Key	<xs:keyref name = "RefKeyDepartmentPK" refer = "KeyDepartmentPK"> <xs:selector xpath = "Employee/EmployeeRow"/> <xs:field xpath = "Dept"/> </xs:keyref> <xs:key name = "KeyDepartmentPK"> <xs:selector xpath = "Department/DepartmentRow"/> <xs:field xpath = "@deptid"/> </xs:key>
Index	Index	<xs:element name = "Employee_Employee_ID_idx"> <xs:complexType> <xs:sequence> <xs:element name="employeeid" type="xs:decimal"/> </xs:sequence> </xs:complexType> </xs:element>

된 각 참조 정보를 이용하여 트리를 생성하는 트리 생성기 부분이다. 셋째, 생성된 트리 정보와 정보 저장소에 존재하는 타입 매핑 정보를 이용하여 메타데이터를 생성하는 부분이다. 넷째, XML 인스턴스를 분석하여 관계 데이터베이스에 데이터를 저장하는 부분으로 구성된다.

- Type Distributor : XML Schema를 파싱하여 엘리먼트 및 어트리뷰트를 분리하고 엘리먼트 처리기 및 어트리뷰트 처리기로 전달한다.
- Element Processor : 엘리먼트의 형식에 따라 Form Processor 내의 복합 엘리먼트 처리기와 단순 엘리먼트 처리기에 전달한다.
- Attribute Processor : 어트리뷰트를 분석하여 전역/지역 여부와 명칭, 그룹 명칭, 데이터 타입

및 출현 여부 등의 정보를 정보 저장소에 저장한다. 표 5는 어트리뷰트 참조 정보의 구조이다.

표 5. 어트리뷰트 참조 정보의 구조

항 목	설 명
구 분	해당 속성이 전역, 지역인지 나타낸다.
어트리뷰트명	해당 속성의 명칭
엘리먼트명 및 그룹명	해당 속성을 포함하는 요소 및 속성 그룹의 명칭
타 입	해당 속성의 데이터 타입을 나타낸다.
USE	해당 속성의 출현 여부를 나타낸다.
참조여부	해당 속성이 전역으로 선언된 속성을 참조하는지 여부를 나타낸다.
참조대상 어트리뷰트명	해당 속성이 참조하는 속성 및 속성그룹의 명칭

표 6. 각 참조 정보의 구조

<엘리먼트 참조 정보 구조>

항 목	설 명
구 분	해당 요소가 루트, 전역, 지역인지 나타낸다.
엘리먼트명	해당 요소의 명칭
부모 엘리먼트명	해당 요소를 포함하는 부모 요소의 명칭
타입	해당 요소가 복잡(com) 또는 단순(sim)형식 인지를 나타낸다.
출현 지시자	해당 요소의 MaxOccurs 값을 가진다.(1 이상의 경우 'U'란 값을 가진다.)
참조여부	해당 요소가 전역으로 선언된 요소를 참조하는지 여부를 나타낸다.
참조 대상 엘리먼트명	해당 요소가 참조하는 요소 및 요소그룹의 명칭

<컬럼 상세 정보 구조>

항 목	설 명
구 분	해당 요소가 엘리먼트, 어트리뷰트에 대한 구분
요소명	해당 요소의 명칭
타입	해당 요소의 기본 타입
길이	해당 요소의 전체 길이
소수점 자릿수	해당 요소의 소수점이하 자릿수
NULL 유무	해당 요소의 NULL 값 허용 유무
디폴트 값	해당 요소의 디폴트 값

<관계 정의 정보 구조>

항 목	설 명
구 분	요소가 Unique인지 Key와 KeyRef인지 구분
관계명	관계의 명칭
경로	관계의 경로를 나타낸다.
필드	관계가 포함하는 필드의 명칭
참조 관계명	관계가 KeyRef인 경우 참조하는 Key의 명
참조 필드명	관계가 참조하는 필드명

• **Form Processor** : 엘리먼트의 형식을 분석하여 정보 저장소에 있는 엘리먼트 참조정보, 컬럼 상세정보와 관계 정의정보를 저장한다. 엘리먼트 참조 정보는 엘리먼트의 정보를 저장하는 테이블 형태의 공간으로 엘리먼트의 전역 및 지역구분, 엘리먼트 명칭, 부모 엘리먼트가 존재하는 경우 부모 엘리먼트의 명칭, 엘리먼트의 타입, 출현 지시자 정보, 참조 여부 등의 정보를 저장한다. 컬럼 상세 정보는 메타데이터를 XML Schema로 변환 시 컬럼의 상세 정보를 교환하기 위해 단순형식을 사용하므로 그 단순형식을 분석하여 각 요소의 데이터 타입, 전체 길이, 소수점 자릿수, NULL 값 허용 유

무, 디폴트 값 등의 정보를 저장한다. 관계 정의 정보는 Unique 및 Key와 KeyRef 참조 메커니즘에 관한 정보를 저장한다. 표 6은 각 참조 정보의 구조이다.

- **Tree Generator** : 정보 저장소의 각 참조 정보들을 사용하여 트리를 생성한다. 트리 생성 방법은 먼저 엘리먼트 참조 정보에서 <schema> 요소를 읽어드려 트리의 루트 노드로 지정한 다음 엘리먼트 참조 정보와 어트리뷰트 참조 정보에서 루트 노드의 직계자식으로 설정된 요소들을 읽어드려 노드를 생성하고 다시 생성된 노드의 자식 요소로 설정된 요소를 읽어 자식 노드를 생성하는 방법으로 트리를 생성해 간다. 각 노드 생성 시 해당 노드를 부모노드로 설정한 요소가 없는 경우 해당 노드는 말단 노드이므로 컬럼 상세정보에서 노드의 데이터 타입, 전체 길이, 소수점 자릿수, NULL 값 허용 여부, 디폴트 값 등의 정보를 노드의 구조에 저장한다.
- **MetaData Generator** : 생성된 트리와 정보 저장소에 존재하는 타입 매핑 테이블을 이용하여 관계 데이터베이스에 메타데이터를 생성하고 XML 문서 매핑 정보를 정보 저장소에 생성한다.
- **InstanceToData Transformer** : MetaData Generator에서 생성한 XML 문서 매핑 테이블을 참조하여 XML Instance의 각 엘리먼트와 어트리뷰트 값을 삽입하기 위한 표준 SQL 스크립트를 작성하여 관계 데이터베이스에 데이터를 저장한다.

그림 5는 XMLToRDB 변환의 각 구성요소들의 처리를 통해 RDB의 객체를 생성하는 과정이다.

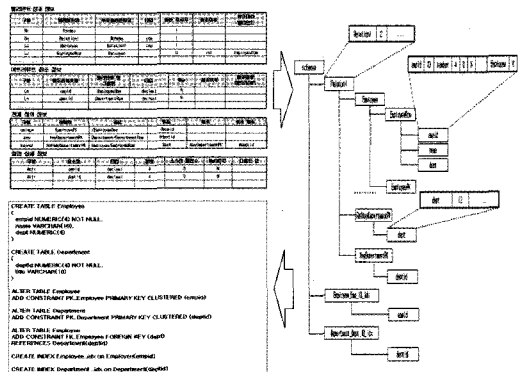


그림 5. XMLToRDB 변환의 각 구성요소들의 처리 과정의 예

표 7. UDDI 각 엘리먼트의 용도 정의

엘리먼트 이름	규정된 용도	본 논문에서 사용될 용도
<businessEntity>	회사에 대한 정보(이름, 주소, 전화번호 등)	회사 내에 구축된 업무용 시스템의 정보
<publisher Assertion>	businessEntity 간의 연관 관계	사용안함
<identifierBag>	businessEntity에 대한 대체 식별자로 사용되는 정보	사용안함
<categoryBag>	분류에 대한 정보	각 업무용 시스템의 분류 정보(예, 재무, 인사, 건설, 자제 등)
<businessService>	회사에서 제공하는 웹 서비스의 이름 및 설명을 기술	업무용 시스템 내의 세부단위 시스템의 정보
<bindingTemplate>	+웹 서비스에 대한 종점 URL 정보 및 웹 서비스와 관련된 tModel을 참조하는 내용 기술	각 단위 시스템의 종점 URL 정보 및 관련된 tModel 참조 정보
<tModel>	웹 서비스에 대한 메소드의 종류 및 인자의 데이터 타입이 정의된 WSDL 문서의 URL 경로 기술	메소드의 종류 및 인자의 데이터 타입이 정의된 WSDL 문서의 URL 정보

3.2 UDDI 레지스트리

사실 UDDI 레지스트리는 회사 내부의 정보를 보호 및 관리하고 정보의 신뢰성을 높이기 위해 사용되며 본 논문에서는 UDDI 스펙 버전 2.0을 따르는 UDDI 레지스트리 정보 저장소를 설계하고자 한다.

UDDI 레지스트리의 데이터 구조는 XML Schema 문서로 정의되고 한 개의 비즈니스 정보를 생성하는데 필요한 일곱 가지의 엘리먼트를 정의하고 있으며 다음 표 7은 각 엘리먼트가 본 논문에서 사용될 용도를 나타낸 것이다.

본 논문에서 UDDI 레지스트리의 저장소로 관계 데이터베이스를 사용하며 위의 표에서 정의된 엘리먼트의 값을 저장하기 위한 관계 테이블 관계도는 그림 6과 같다.

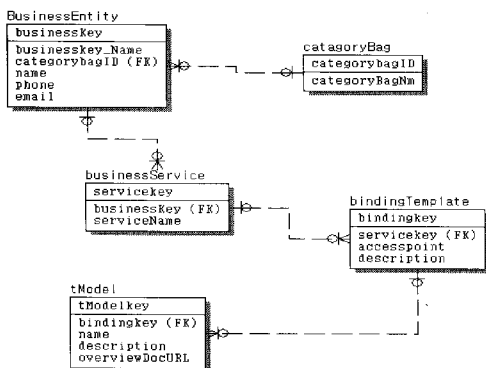


그림 6. 테이블 관계도

그림 7은 위의 관계도를 바탕으로 구성된 UDDI 레지스트리 저장소를 위한 UDDI Server 시스템 구조이다. UDDI Server는 SOAP 메시지에 XML 문서를 포함하거나 추출하는 부분과 XML 문서를 파

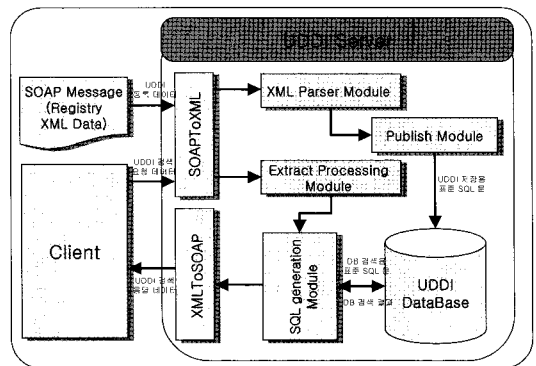


그림 7. UDDI Server System 구조

싱하여 정보를 얻어내는 부분, 얻어진 정보를 관계 데이터베이스에 저장하거나 XML 문서를 생성하는 부분으로 구성되어 있다.

- SOAPToXML과 XMLToSOAP : SOAP 메시지에 포함되어 전송된 XML 문서를 추출하고 응답 결과를 포함한 XML 문서를 SOAP 메시지에 포함하여 전송한다.
- XML Parser Module : 전송된 XML 문서를 파싱하여 UDDI 데이터의 구조를 분석하고 결과를 Publish Module에 전달한다.
- Publish Module : 분석된 UDDI 데이터를 데이터베이스에 저장하기 위한 표준 SQL문을 생성하고 데이터를 저장한다.
- Extract Processing Module : 전송된 XML 문서를 파싱하여 검색 조건을 추출한다.
- SQL generation Module : 추출된 검색 조건으로 표준 SQL 문을 작성하고 결과를 XML 문서로 생성한다.

3.3 웹 서비스

웹 서비스는 웹 서버, 웹 서비스 시스템, 비즈니스 서버로 구성된다. 웹 서버는 원격 프로시저의 위치정보 및 매개변수, 리턴 변수 형식 등의 정보를 포함하고 있는 WSDL 문서를 저장/관리하고 웹 서비스 시스템은 원격 프로시저의 인터페이스를 관리하고 클라이언트의 요청이 있는 경우 요구하는 프로시저를 실행하여 결과를 반환 하는 역할을 수행한다. 비즈니스 서버는 실질적인 작업을 수행하는 원격 프로시저를 관리하고 실행하는 역할을 수행하고 또한 웹 서비스의 클라이언트로써 자체적으로 필요한 서비스를 웹 서비스에 요청하여 얻은 결과를 처리하여 자체 데이터베이스에 반영한다.

웹 서버는 제공자가 원격 프로시저의 명, 인자, 반환형, SOAP 메시지의 전송 프로토콜의 종류, 웹 서비스 시스템의 URL 등의 정보를 명세화 하여 저장한 WSDL 문서를 저장/관리하고 UDDI 레지스트리에 WSDL 문서를 다운 받을 수 있는 URL를 공개하며 소비자가 UDDI 레지스트리를 검색하여 필요로 하는 WSDL문서를 요청하는 경우 해당 WSDL 문서를 다운로드할 수 있도록 하는 역할을 수행한다. 웹 서버는 WSDL 문서의 URL 정보를 포함하는 XML 문서를 UDDI 스펙에 맞게 작성하여 SOAP 메시지에 포함하고 UDDI Server System으로 전송한다. 처리 과정은 그림 8과 같다.

웹 서비스 시스템은 비즈니스 서버 상에 존재하는 실질적인 프로시저들의 인터페이스들을 구현한 원격 프로시저들과 요청 SOAP 메시지의 디코딩 작업을 수행하는 타이 클래스가 위치한다. 또한 WSDL 문서를 생성하여 웹 서버에 저장한다. 전송용 프로토콜은 H/W나 S/W에 상관없이 요청 SOAP 메시

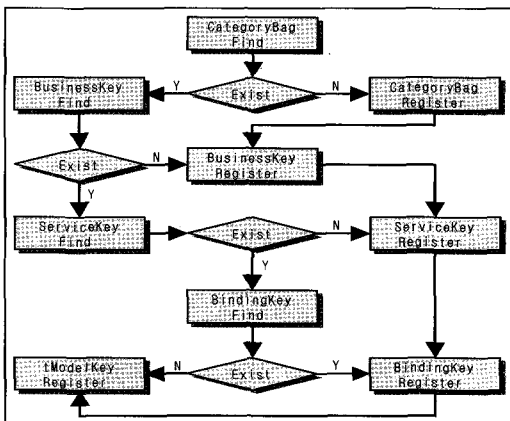


그림 8. 웹 서비스 공개 처리 순서도

지를 보내고 응답 SOAP 메시지를 받아 처리할 수 있는 HTTP 프로토콜을 사용한다. 비즈니스 서버는 웹 서비스 클라이언트로써의 역할이고 다른 하나는 웹 서비스 시스템의 실질적인 프로시저를 포함하고 있는 서버로써의 역할이다. 전자는 다른 비즈니스 서버에서 데이터를 가져오기 위해 UDDI 레지스트리를 검색하여 WSDL 문서의 URL 정보를 받아 웹 서버로부터 WSDL 문서를 다운받은 다음 WSDL 문서 처리기를 통해 처리하여 클라이언트 클래스를 작성하고 클라이언트 객체를 생성하여 원격 프로시저명과 인자 정보를 요청 SOAP 메시지로 인코딩하는 스텝객체를 통해 웹 서비스에 요청 SOAP 메시지를 보낸다. 스텝객체는 응답 SOAP 메시지를 디코딩하여 클라이언트 객체를 통해 클라이언트 프로시저가 결과를 받아 데이터를 저장한다.

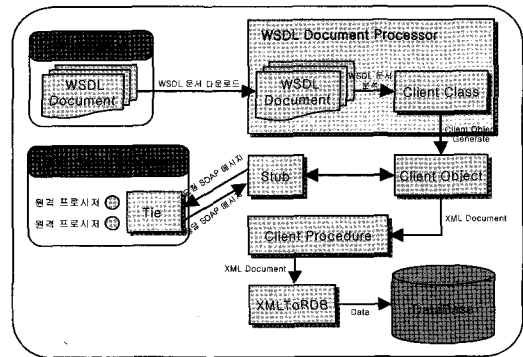


그림 9. 웹 클라이언트 처리과정

후자는 웹 서비스의 원격 인터페이스가 실행할 실질적인 프로시저를 개발하여 배포하는 것으로 생성하는 프로시저는 의미적인 제약 조건을 포함하는 메타데이터 및 데이터를 읽어들이 XML Schema, XML Instance 문서를 생성하는 컴포넌트를 포함하며 이렇게 생성된 XML 문서를 원격지의 웹 클라이언트에게 제공한다. 그림 9는 웹 클라이언트의 처리과정을 나타낸다.

IV. 웹 서비스를 이용한 이기종 RDB간의 데이터 이주 시스템 구현 및 테스트

웹 서비스를 기반으로 기업 내에 분산되어 존재하는 데이터를 이주하는 데이터 이주 시스템을 구현하고 테스트한 환경은 표 8과 같다. DBMS는 오라클 9i 및 SQL Server 2000을 사용하고 데이터의 교환은 XML Schema 표준을 사용한다. 웹 서비스

표 8. 시스템 구현 및 테스트 환경

운영체제	Windows 2000 professional
서블릿 엔진	Tomcat 4.1.3
DBMS	Oracle 9i, MS SQL Server 2000
웹 서비스 개발 도구	JWSDP
구현 언어	JAVA, JSP, EJB
사용 API	JAXP, JAX-RPC, JAXB, SAAJ
전송 프로토콜	SOAP
DB 연결	ODBC:JDBC

시스템을 구현하기 위해 JAVA에서 제공하는 웹 서비스 시스템 통합 개발 킷인 JWSDP를 사용한다. WSDL 문서를 저장 및 관리하고 UDDI 레지스트리에 WSDL의 URL를 등록하는 웹 서버와 웹 서비스를 제공하는 웹 서버 및 서블릿 엔진으로는 Apache Group의 Tomcat 4.1.3을 사용한다. XML과 RDB간의 변환 컴포넌트는 EJB로 작성한다.

4.1 XML과 RDB간의 변환 컴포넌트 구현

XML과 RDB 상호간의 변환 작업을 수행하는 컴포넌트인 XR-InTrans는 RDBToXML과 XMLToRDB 부분으로 구성되어 있다.

RDBToXML은 EJB의 상태 없는 빈으로 구현되며 의미적인 제약 조건을 포함한 메타데이터와 데이터를 Java API에서 지원하는 메타데이터 추출 인터페이스 및 JDBC API를 사용하여 추출하고 이 정보를 XML Schema와 XML 인스턴스로 생성하는 컴포넌트이다. 본 논문에서는 테이블의 메타데이터인 칼럼 및 키, 인덱스, Not Null 제약 조건 등의 정보와 실제 데이터만을 이주 대상으로 한다.

RDBToXMLEJB는 상태 없는 빈 클래스로 실질적인 변환 작업을 수행하는 각 클래스들을 호출하며, RDBToXMLHome은 홈 인터페이스로 RDBToXML 객체를 생성하는 역할을 수행하고 RDBToXML은 원격 인터페이스이다. 그림 10은 RDBToXML의 순

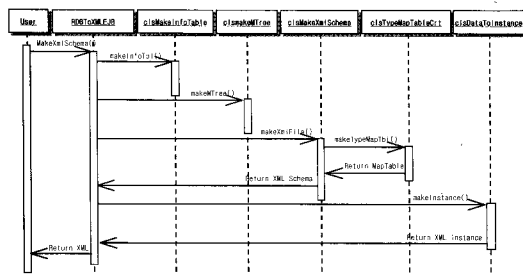


그림 10. RDBToXML 순차 다이어그램

차 다이어그램을 나타낸 것으로 'clsMetaInfoTable' 클래스는 테이블들의 메타데이터를 추출하여 각 정보 저장소에 컬럼, 인덱스, 제약 조건 정보 등을 저장하고 'clsMakeXMLTree' 클래스는 정보 저장소의 정보를 활용하여 트리를 생성한다. 'clsXMLSchema' 클래스는 트리와 타입 매핑 정보를 이용하여 XML Schema 문서를 작성하며 'clsDataToInstance' 클래스는 테이블의 실제 데이터를 이용하여 XML 인스턴스 문서 작성한다.

XMLToRDB 변환 컴포넌트는 XML Schema와 XML 인스턴스를 분석하여 관계 데이터베이스의 의미적인 제약 조건을 포함하는 메타데이터를 생성하고 데이터를 저장하는 컴포넌트이다. 파싱된 XML Schema 정보를 정보 저장소에 존재하는 각 참조 테이블들에 저장하고 이를 사용하여 트리를 생성한다. 이렇게 생성한 트리과 타입 매핑 정보를 사용하여 메타데이터를 생성하고 메타데이터 생성 시 저장한 XML 문서 매핑 정보를 참고하여 XML 인스턴스로 연계된 실제 데이터를 관계 데이터베이스에 저장한다.

XMLToRDBEJB는 상태 없는 빈 클래스로 실질적인 변환 작업을 수행하는 각 클래스들을 호출하며, XMLToRDBHome은 홈 인터페이스로 XMLToRDB 객체를 생성하는 역할을 수행하고 XMLToRDB는 원격 인터페이스이다. 그림 11은 XMLToRDB의 순차 다이어그램을 나타낸 것으로 'clsMakeRefTable' 클래스는 XML Schema를 파싱하여 정보 저장소에 존재하는 각 참조 구조에 정보를 생성하며 이렇게 생성한 정보를 이용하여 'clsCRTxmlTree' 클래스는 트리를 생성한다. 'clsCRTMateData' 클래스는 트리 정보와 컬럼 상세 정보를 이용하여 메타데이터 생성용 스크립트를 작성하여 타겟 데이터베이스에 메타데이터를 생성한다. 'clsInstanceToData' 클래스는 XML 인스턴스를 분석하여 실제 데이터를 이미 생성된 테이블에 저장한다.

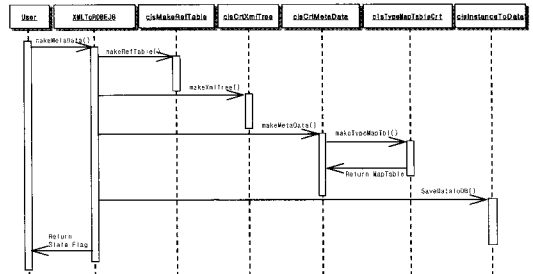


그림 11. XMLToRDB 순차 다이어그램

4.2 웹 서비스 시스템 구현

본 논문에서 구현하는 UDDI 레지스트리는 RegServiceComp 및 FindServiceComp 컴포넌트로 구성된다. RegServiceComp 컴포넌트는 제공자가 웹 서비스를 등록하기 위해 전송한 SOAP 메시지를 해석하여 XML 문서를 분리하고 분리된 XML 문서를 파싱하여 등록할 자료를 UDDI의 데이터베이스에 저장한다. FindServiceComp 컴포넌트는 웹 서비스 소비자가 필요로 하는 웹 서비스를 찾고자 UDDI 레지스트리를 검색하기 위해 전송한 SOAP 메시지를 해석하여 요청된 검색 결과를 소비자에게 전송하는 역할을 한다. 그림 12는 UDDI 레지스트리에 정보를 등록하고 검색하는 과정을 나타내는 순차 다이어그램이다.

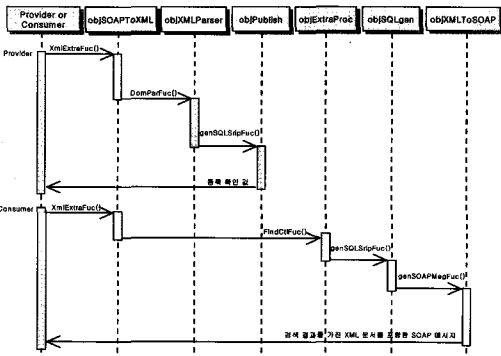


그림 12. UDDI 레지스트리 등록 및 검색 순차 다이어그램

그림 13은 UDDI 레지스트리에 정보를 검색하는 화면으로 메인 화면의 좌측에 분류 정보가 나타나고 우측에 businessEntity 정보가 나타난다. 우측의 businessEntity 항목 중 하나를 선택하면 그림과 같이 BusinessService 검색 화면이 나타난다. 각 화면에 나타나는 항목을 클릭하면 해당 항목에 종속되는 하위항목을 보이는 화면으로 이동한다. 각 businessEntity, businessService, bindingTemplate, tModel 화면의 하단에 있는 등록버튼을 클릭하며 그림 13에서와 같이 새로운 서비스를 UDDI에 등록할 수 있는 화면으로 이동한다.

그림 14는 UDDI 레지스트리에 정보를 등록하는 과정을 나타내는 화면으로 각 검색화면에서 등록버튼을 클릭한 경우 나타나며 그림에서와 같이 Business Service 등록 시에는 Service Key와 Service Name를 등록하고 Binding Template 등록 시에는 Binding Key 및 서비스의 Access Point인 URL를 등록한다.

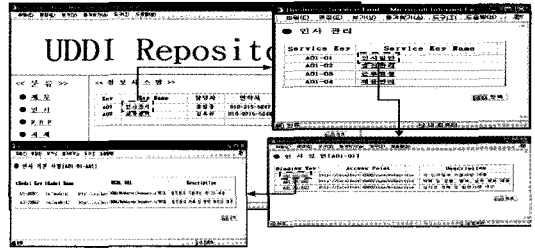


그림 13. UDDI 레지스트리 검색

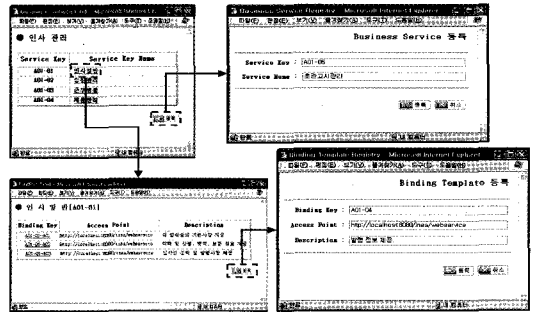


그림 14. UDDI 레지스트리 등록

웹 서비스 시스템은 비즈니스 서버의 실질적인 프로시저와 RMI로 연결되어 호출할 수 있는 인터페이스 및 구현 클래스를 생성하고 배치 기술자와 RMI 인터페이스 배치 기술자를 작성한 다음 해당 서비스의 타이 클래스와 WSDL 문서를 생성한다. 생성한 WSDL 문서는 웹 서버에 저장한다. 다음은 원격 프로시저를 호출하기 위한 SOAP 메시지의 예이다.

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:enc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns0="http://localhost:8080/wsdl/webservice"
  env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <env:Body>
    <ns0:SelectEmployee>
      <int_1 xsi:type="xsd:int">1004</int_1>
      <String_2 xsi:type="xsd:string">홍길동</String_2>
      <int_3 xsi:type="xsd:int">7755</int_3>
    </ns0:SelectEmployee>
  </env:Body>
</env:Envelope>
    
```

그림 15. 웹 프로시저 호출용 SOAP 메시지

웹 서버는 웹 서비스 시스템에서 작성된 WSDL 문서를 저장하고 WSDL 문서의 위치를 구현한 UDDI 레지스트리에 등록하기 위한 XML 문서를 SOAP 메시지를 통해 UDDI 레지스트리에 전송한다. 다음은 웹 서버에 저장된 WSDL 문서의 예이다.

다음 그림 17과 18은 위에서 구현 및 테스트한 데이터 이주 시스템을 적용한 예를 나타낸다. 오라클을 기반 RDBMS로 사용하는 인사 시스템의 데이

V. 비교분석 및 결론

본 논문에서는 인트라넷 상에서 웹 서비스를 이용한 관계 데이터베이스간의 데이터 이주를 위한 시스템을 설계하고 테스트하였다. 이주 대상으로는 테이블의 컬럼 정보와 의미적인 제약 조건인 인덱스, 기본 키와 외래 키 정보, Not Null 조건 등의 정보와 실 데이터이다.

XR-InTrans 컴포넌트를 사용하여 XML 문서와 관계 테이블간의 변환을 수행하고 구현된 UDDI 레지스트리에 원격 프로시저의 정보를 저장하는 WSDL 문서의 URL 정보를 각 업무별로 저장하여 해당 업무의 데이터를 필요로 하는 시스템에서 접근하여 데이터를 제공하는 원격 프로시저의 명 및 인자, 반환형 등의 정보를 획득할 수 있도록 하였다. 본 논문에서 구현한 UDDI 레지스트리는 XML 형태의 UDDI 데이터를 관계 데이터베이스에 효율적으로 저장 및 검색할 수 있는 구조로 설계하였다. 본 논문에서 설계하고 테스트한 시스템은 기존의 EAI와 같은 환경에서도 가능하나 EAI의 단점을 보완할 수 있다. 다음은 EAI 및 타 논문^[8]과 본 논문에서 구현한 시스템간의 비교는 표 9와 같다.

표 9를 보면 EAI에서의 일반적인 데이터를 교환하기 위한 방법으로 원격지에 위치한 RDB의 특정 테이블을 복제하는 DBToDB 방식과 프로시저를 생성하여 교환하는 방식이 있다. 또한 비교 대상 논문^[8]은 XML 문서의 데이터를 교환하기 위한 방법으로 XMLToXML 방식을 사용한다. 반면 본 논문에서는 RDB의 데이터를 이주하기 위해 RDB의 데이터를 XML문서 형식으로 변환하여 교환하는 RDBToRDB 방식을 사용한다. 이와 같이 본 논문은 EAI의 고정된 연결 및 연결 대상 정의, 고정적인 관리자 필요 등의 특성을 비교 대상 논문^[8]과 같이 유동적이며 연결 대상의 미리 정의할 필요가 없는 특성을 취하고 있어 원격지에 존재하는 이기종 RDB간의 데이터를 시스템 환경에 구애받지 않고 이주할 수 있다.

표 9의 비교에 따라 본 논문에서 구현한 시스템의 데이터 이주 성능을 알아보기 위해 표 10과 같은 테스트 환경 및 방법으로 EAI와 본 논문을 비교한다.

그림 19는 이주 대상으로 특정한 하나의 테이블을 선정하고 해당 테이블의 실제 데이터(ROW Size : 115 Byte) 건수가 10000, 30000, 50000일 때 데이터를 이주하는데 걸리는 시간을 비교한 것이다.



그림 16. 웹 서비스에서 작성된 WSDL 문서

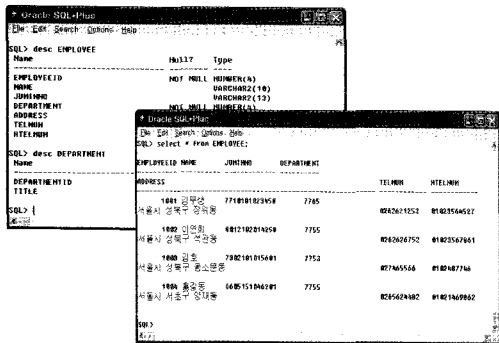


그림 17. 오라클에 저장된 원본 데이터

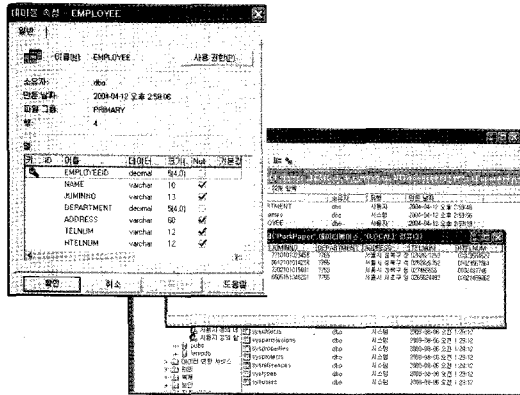


그림 18. MS SQL Server로 이주된 데이터

터를 MS SQL Server를 RDBMS로 사용하는 타 시스템에 제공한 결과로 본 논문에서 설계하고 테스트한 데이터 이주 시스템은 변환 표준으로 XML Schema를 사용하고 전송표준으로 SOAP 및 HTTP 프로토콜을 사용하기 때문에 이 기종 RDBMS 간의 데이터 이주도 가능하게 한다.

표 9. EAI 및 타 논문^[8]과 본 논문의 비교

	EAI	타 논문 ^[8]	본 논문
구현	다양한 기술 요구	SOAP	SOAP
표준	J2CA	SOAP, UDDI, WSDL	SOAP, UDDI, WSDL
교환 방법	DBToDB : 원격지에 위치한 RDB의 특정 테이블 복제 RPC : EAI 내부에 프로시저를 생성하여 데이터 제공	XMLtoXML : 시스템간의 XML 데이터를 이주하기 위한 기술을 활용하여 전송한다.	RDBtoRDB : 각 비즈니스 서버에 데이터를 제공할 원격 프로시저를 생성하여 XML 형태로 데이터를 이주한다.
연결 방법	고정된 연결 필요 연결대상이 미리 결정되어 있어야함.	유동적인 연결. 연결대상이 미리 정의되어 있지 않음.	유동적인 연결 가능. 연결대상이 미리 정의되어 있지 않고 필요시에만 연결이 가능
유지 관리	고정적인 관리자에 의해 관리 되어져야 함.	고정적인 관리자에 의해 관리 되어짐.	고정적인 관리자가 필요하지 않음

표 10. 테스트 환경 및 방법

테스트 환경	OS	Microsoft Windows 2000 5.00.2195	
	CPU / RAM	Intel Pentium 4 2.00GHz / 512MB	
	RDBMS	Microsoft SQL Server 2000 8.00.194 Oracle 9i	
테스트 방법	데이터 건수	10000, 30000, 50000	
	교환 방법	EAI	DBToDB
		EAI	RPC
		본 논문	웹 서비스를 이용한 데이터 이주
	방법	1. 이주 대상 테이블의 스키마 및 실제 데이터를 소스 RDB에 타겟 RDB로 이주하여 대상 테이블을 생성하고 실제 데이터를 저장하는데 걸리는 시간	

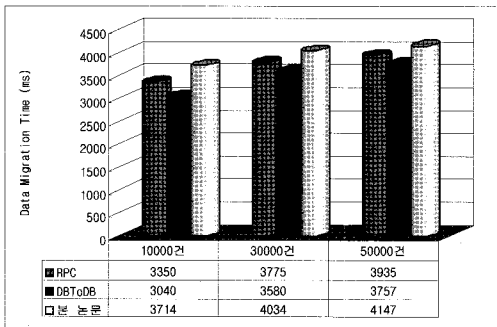


그림 19. 데이터 이주 측정 결과

위의 측정 결과를 보면 기존의 EAI가 본 논문에서 구현한 시스템보다 데이터를 이주하는데 드는 시간은 적게 걸리나 EAI에서는 이주 대상인 테이블을 타겟 RDB에 미리 생성해야 하는 등의 추가적인 수작업을 요구한다. 그러나 본 논문에서 구현한 시스템은 이주 대상인 테이블의 스키마를 이주하여 타겟 RDB에 생성하고 생성된 테이블에 이주된 실제 데이터를 저장하므로 추가적으로 수행하여야 하는 수작업의 양을 줄일 수 있고 이주되는 데이터의 양이 증가할수록 이주 시간의 차이가 줄어드는 것을 알 수 있다.

본 논문에서 구현된 XR-InTrans 컴포넌트는 기존의 논문의 XML 문서와 RDB간의 변환과는 다음과 같은 차이를 갖는다.

- 1) 테이블의 컬럼 정보 및 의미적인 제약 조건인 인덱스, 키, Not Null 조건 등의 메타데이터를 이주 대상으로 한다.
- 2) 본 논문은 eXcelon Server와 같은 독립적인 저장 공간이 필요치 않다.
- 3) 이주 대상은 테이블(Table) 뿐만 아니라 데이터를 보호하고 보안을 유지하기 위해 사용하고 있는 뷰(View)도 포함한다.
- 4) 이주되는 데이터 타입의 상세 정보를 전달할 수 있다.

현재 기업들은 기업 내에 구축된 각각의 정보 시스템으로부터 대량의 정보들을 생산하고 있으며 이러한 정보들을 저장하기 위해 관계 데이터베이스를 사용하고 있다. 이렇게 분산되어 저장된 정보들을 효율적으로 이용하기 위해서는 데이터 이주가 필요하며 기업 내에 구축된 Intranet상에서 운영이 가능한 웹 서비스를 사용함으로써 특정 솔루션에 의존적인 EAI의 단점을 보완할 수 있고 구축 및 유지 비용을 절감할 수 있다. 본 논문에서는 웹 서비스를 이용해 각 업무 시스템 내에 위치한 관계 데이터베이스

이스간의 메타데이터 및 데이터 이주를 위한 컴포넌트 및 시스템을 설계하고 테스트하였다.

향후에는 구현된 UDDI 레지스트리에 보다 효율적으로 정보를 저장하고 검색할 수 있도록 정보 개시 및 검색(Information Publish & Retrieval) 기능을 사용하는 방법을 새로운 UDDI 3.0 스펙을 기반으로 하여 연구하고자 하며 웹 서비스를 이용해 교환되는 데이터의 동기화 기능, 데이터 보안 기능에 관해 연구하고자 한다.

참 고 문 헌

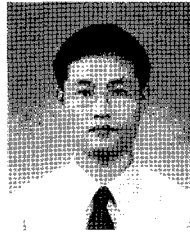
- [1] Heather Kreger, "Web Services Conceptual Architecture(WSCA 1.0)", IBM Software Group, May 2001.
- [2] Gerti Kappel, Elisabeth Kapsammer & Werner Retschitzegger, "Integrating XML and Relational Database Systems", WWW:Internet and Web Information Systems, 7, 343-384, 2004
- [3] 신민철, "XML 웹 서비스", FREELEC, 2004
- [4] W3C, "Web Services Architecture, W3C Working Draft", May 14, 2003
- [5] UDDI.org, "UDDI Technical White Paper", September 2000
- [6] Roberto Chinnici, "Web Service Description Language(WSDL) Version 2.0 Part 1 : Core Language", W3C Working Draft 3, August 2004
- [7] Nilo Mitra, "SOAP Version 1.2 Part 0 : Primer", W3C Recommendation, June 2003
- [8] Sihem Amer-Yahia, Yannis Kotidis, "A Web-Services Architecture for Efficient XML Data Exchange", ICDE'04 IEEE, 2004
- [9] Bei Jia, Cai Fei, Tao Lie-Jun, Pan Jin-Gui, "A Direct Method of Data Exchange between XML and Relation Database", ITI 2004, June 7-10 2004
- [10] 신경운, "사상규칙을 이용한 XML 문서 변환기의 설계", 충북대학교 대학원 석사학위논문, 2004.2
- [11] M.Carey, D.Florescu, Z.Ives, Y.Lu, J.Shanmugasun-daram, E.Shehita, and S.Subramanian, "XPERANTO : Publishing Object-Relation Data as XML" Workshop on the Web and Databases (WebDB), pp. 105-110, May, 2000
- [12] R.Bourret, C.Bornhovd, A.P.Buchma nn, "A Generic Load/Extract Utility for Data Transfer between XML Documents and Relation Databases" WECWIS'00, San Jose, California,

Jun.8-9, 2000

- [13] 조정길, "XML 문서의 관계형 스키마 변환 기법", 충북대학교 대학원 박사학위 논문, 2003.2
- [14] 김정섭, "XML Schema로부터 관계형 스키마의 자동 생성", 한국과학기술원 석사학위논문, 2002
- [15] 최우태, "XML을 기반으로 한 메타데이터 리프 지토리 설계 및 구현", 숭실대학교 대학원 석사학위논문, 2003.12

박 유 신 (Yoo-Shin Park)

정회원



2002년 2월 한성대학교 컴퓨터 공학과 졸업
2004년 8월 광운대학교 정보통신대학원 컴퓨터공학과 석사
1998년 4월~2005년 9월 (주)엠씨에스텍
2005년 10월~현재 근로복지공단 정보시스템실

<관심분야> 데이터통합, 그리드컴퓨팅, BPEL, 웹 서비스 XML

정 계 동 (Kye-Dong Jung)

정회원



1985년 광운대학교 전자계산학과 졸업(학사)
1992년 광운대학교 산업정보학과 석사
2000년 광운대학교 컴퓨터과학과 박사
1993년~2004년 광운대학교 정보과학원 교수

2004년~현재 광운대학교 교양학부 교수

<관심분야> XML 분산시스템, 분산컴퓨팅기술, 이동 에이전트

최 영 근 (Young-Keun Choi)

정회원



1980년 서울대학교 수학교육과 졸업(이학사)
1982년 서울대학교 계산통계학과 이학석사
1989년 서울대학교 계산통계학과 이학박사
1983년~현재 광운대학교 컴퓨터과학과 교수

1992년~2000년 광운대학교 전산정보원 원장

2002년~2004년 광운대학교 교무연구처장

<관심분야> 프로그래밍언어, 분산시스템, 이동에이전트, 상호운용성