

ASP 용의 데이터베이스와 고객 데이터베이스 연동에 관한 연구

김 호 윤* · 이 재 원**

요 약

ASP(Application Service Provider) 사업에서 응용프로그램이 데이터베이스를 사용해야 하고 이 데이터베이스가 ASP 고객의 데이터베이스로부터 일련의 데이터를 제공받아야 하는 경우가 점점 더 증가하고 있다. 이를 위한 방법으로 엑셀(EXCEL)과 같은 파일 형태가 이용되는데 고객의 데이터베이스로부터 수작업으로 추출되어진 데이터가 엑셀 파일로 ASP 사업자에게 전달된다. 이는 다시 애플리케이션이 이용하는 데이터베이스의 테이블내로 수작업 처리되어 저장된다. 본 연구는 두 데이터베이스 연동 시에 이러한 엑셀 파일 등을 이용하여 발생하는 데이터 추출과 저장시의 데이터베이스 수작업 조작의 비 효율성을 극복하는데 있다. 이 때 두 데이터베이스 상의 속성명은 상이한 경우를 일반적인 경우로 전제하였으며 또한 웹 상에서 두 데이터베이스의 연동 시에 피할 수 없는 고객 데이터베이스의 경로 정보를 ASP 사업자에게 노출을 피하도록 한다. 제안된 방법은 두 단계로 걸쳐 데이터 프로세싱이 이루어지는데 먼저, ASP 사이트로부터 다운로드 받은 프로그램을 이용하여, 고객 데이터베이스를 접속한 후 프로그램의 유저 인터페이스를 이용하여 용이하게 속성 매칭하고 속성치를 XML 파일 형태로 자동 추출하는 단계와 추출된 XML 파일을 웹 상에서 ASP 서버에 접속하여 전송하고 ASP 데이터베이스에 자동 저장하는 후처리 프로세싱 단계로 구성된다. 본 연구에서는 이러한 방법에 맞는 프로토타입 시스템을 구현하여 데이터베이스 연동의 타당성을 검증하였으며, 제안된 방법은 ASP 고객의 데이터베이스를 이용해야 하는 ASP 사업의 활성화에 기여할 수 있다.

A Study of Integrating ASP Databases with Customer Databases

Ho-Yoon Kim* · Jae-Won Lee**

ABSTRACT

In the ASP(Application Service Provider) business, applications using database sometimes require some data from clients' databases. These days such data are extracted from client database using manual database operations as an EXCEL file and the ASP, once receiving this file, transfers it into the application's database using manual database operations. This paper describes how to deal with data transmitting between the client database and ASP database on the web without using database manual operations for data extraction and insertion. We propose a framework which enables to transmit client data in a systematical way, to match different attribute names of each database for sharing same attribute values, and to avoid exposing information about the network path of client database to the ASP. This approach consists of two steps of data processing. The first is extracting data from client database as XML format by using a downloaded client program from ASP site, the second is uploading and storing the XML file into the ASP database. The implemented prototype system shows the suggested data integration paradigm is valid and ASP business needing integration of client database can be activated using it.

키워드 : ASP, 데이터베이스(Database), XML, 데이터베이스 연동(Data Transmission), 속성 매칭(Attribute Matching)

1. 서 론

최근 들어 인터넷 기술의 발전으로 기업 내에 보유하고 있는 정보의 수집, 공유, 분석에 얼마나 효율적으로 정보기술을 사용하는가에 많은 관심을 가지고 있다. 그 예로서, 신규 프로젝트의 일환으로 인터넷 기반의 고객관계관리(e-

CRM), ERP(Enterprise Resource Planning) 시스템 및 데이터웨어하우스(datawarehouse) 시스템 등의 도입이 추진되어 질 때에 이들 시스템이 갖고 있는 독자적인 데이터베이스와 기존의 기업이 사용하고 있던 기간시스템의 데이터베이스와의 통합 문제가 발생할 수 있다. 특히, 독자적인 데이터베이스를 갖는 응용프로그램을 웹 서버(Web Server) 상에서 ASP(application service provider) 응용프로그램 형태로 제공하는 경우 ASP 고객 데이터베이스와의 데이터 공유를 위한 연동의 필요성이 증가되고 있다.

* 이 논문은 2002년도 인하대학교의 지원에 의하여 연구되었음(INHA-22671)

† 준 회원 : 인하대학교 대학원 첨단정밀공학과

** 정 회원 : 인하대학교 기계공학과 산업공학전공 교수
논문접수 : 2004년 2월 4일, 심사완료 : 2004년 6월 25일

ASP 사업은 정보기술 수요의 일부 또는 전부를 아웃싱 하려는 회사들을 위한 방안이다. ASP사업의 특성은 소프트웨어 기반의 서비스 및 솔루션들을 중앙 데이터 센터로부터 인터넷망을 통해 고객들에게 제공하는 서비스이다[12]. 기업이 ASP 서비스를 임대하는 경우 ASP 측의 데이터베이스로 웹 상에서 기업 내부의 데이터 전송을 필요로 한다.

국내의 ASP 사업자의 데이터베이스와 ASP 고객의 데이터베이스 연동에 관한 현황은 아직 웹상에서 시스템적으로 데이터 전송처리가 드물며, ASP 고객 자신이 사용하고 있는 데이터베이스에서 엑셀(Excel) 등과 같은 파일 형태로 데이터를 추출하여 ASP 사업자에게 전달하고, ASP 사업자는 이를 ASP사이트의 데이터베이스에 수작업으로 입력하는 방법이 이용되고 있다.

본 연구 목적은 이러한 수작업 데이터 전달의 비효율성을 극복하고 고객 데이터베이스 자료를 웹 상에서 ASP 서버 측으로 직접 전송할 수 있는 프레임워크를 제시하는 것이다. 고객의 데이터베이스가 ASP 서버 측과의 접속이 이루어지는 과정에서 고객 데이터베이스의 네트워크상 경로가 노출될 경우가 있다. 본 연구에서는 이러한 두 데이터베이스가 웹 상에서 직접 연결로 데이터 전송시 고객 데이터베이스 네트워크 상의 접속경로 정보 노출의 문제를 회피하고, SQL(structured query language)에 관련된 지식이 없는 ASP 임대 사용자도 데이터 전송을 쉽게 처리할 수 있도록 하는 시스템을 제안한다. 제안된 시스템에서 고객은 ASP 웹사이트에서 제공되는 클라이언트 응용프로그램을 이용하여, ASP 웹 서버와는 오프라인(off-line) 상태로 자신의 데이터베이스 서버에 접속한다. 이후 ASP 웹사이트의 데이터베이스가 요구하는 속성명을 매칭 시켜 그 속성값을 XML 파일로 추출한다. 그리고 이를 온라인(on-line)으로 ASP 사이트에 접속, 추출된 XML 파일을 전송하여 데이터베이스에 자동 입력한다. 이에 관한 상세 내용은 3장에서 다루고 있다.

본 논문의 이후는 다음과 같이 구성되어 있다. 2장은 데이터베이스 연동, XML을 이용한 데이터 전송 등에 관한 관련연구를 약술한다. 3장에서는 제안된 시스템에 관한 구체적인 내용은 기술하고 4장에서는 구현된 프로토타입 시스템 D2D를 소개한다.

2. 관련 연구

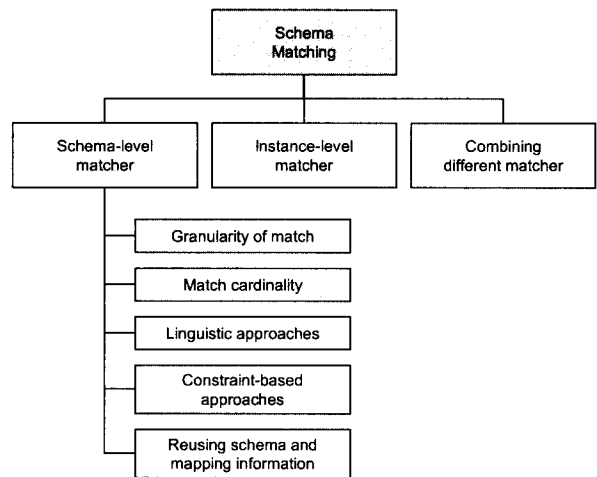
2.1 데이터베이스 연동

데이터베이스 연동은 여러 분야에서 필요로 하고 있으나 본 연구에서는 ASP 서버의 데이터베이스와 ASP 고객의 데이터베이스를 인터넷 상에서 연동 경우를 대상으로 한다. Wiedemann[1]이 ASP사업에서 데이터베이스의 연동의 필

요성을 언급하고 시뮬레이션 모델을 통하여 데이터 연동을 다룬바 있으나, 실제적인 상황에 대한 연동 방법을 제시하지는 않았다. ASP 사업에서의 데이터를 전송 받는 경우에 대한 연구는 아니지만, 임의의 두 데이터베이스의 연동에 관한 연구는 일부 진행 되어 있다. 이 분야의 연구로서 Sheth와 Larson[2]은 DBMS 데이터 모델의 이질성 종류를 분석하였는데, 동일한 속성명이 의미상으로 다른 경우 데이터 연동은 다루었으나, 속성명이 상이하고 동일한 속성 값을 공유해야 하는 것은 다루지 않았는데 이는 본 연구의 관심 부분이다. Lawrence와 Barker[3]은 스키마 매칭을 위한 방법으로 스키마의 엘리먼트 의미를 기술한 표준 용어 사전(standard term dictionary) 이용을 제시하였는데 이는 속성 매칭 테이블을 사용하여 속성 매칭하는 방법과 유사하다. Rahm과 Bernstein[4]은 데이터 모델의 스키마 매칭 방법을 구분하고 각각의 매칭 방법을 응용한 프로토타입 스키마 매치들을 소개 하였고 스키마 매칭에 대해 다음과 같이 정의하였다.

‘매칭이란 두개의 스키마를 입력 값으로 하고, 상호 의미상 대응하는 두 스키마의 엘리먼트들 간의 매핑을 보여주는 것이다.’

이들은 매칭 방법을 스키마 레벨 매치(Schema-level matcher), 인스턴스 레벨 접근법(Instance-level approaches), 그리고 결합 매치(Combining different matchers)로 분류하였다. (그림 1)은 매칭 방법 분류에 대해 도시한 것이다.



(그림 1) 매칭 방법 분류[4]

본 연구에서는 소스 데이터의 일부 속성치를 타겟 데이터베이스로 전송하는 경우이고, 두 데이터베이스의 속성명으로부터만 매칭해도 속성치 전달이 가능하다. 그러므로 스키마 레벨 매치 분류 중 언어적 접근법(linguistic approaches)의 Name 매칭 방법을 이용한다.

이러한 데이터베이스의 매칭을 위한 데이터 접근 도구로

이질적인 분산 데이터베이스상에서 CORBA(Common Object Request Broker Architecture)를 이용하는 연구가 있다. 변광준[8]이 수행한 연구에서는 CORBA를 이용하여 각각의 데이터베이스 사용자가 공유에 필요한 데이터베이스 정보인 데이터베이스명, 테이블명 등을 메타 데이터로 작성하여 서버에 저장하고, 클라이언트는 서버에 저장된 메타 데이터를 이용하여 속성 매칭과 속성치를 공유 하도록 하였다. 그리고 Mark 외[5]은 CORBA구조상에서 멀티데이터베이스 연동을 위해 번역 지식베이스(Translation Knowledge Base)를 구축하고, 이를 이용하여 로컬 데이터베이스의 스키마 연동을 했다. 하지만 CORBA를 이용하여 웹 응용프로그램을 개발할 경우 서버와 클라이언트 사이에 미들웨어가 존재하는 3계층(3-tire) 방식으로 시스템이 개발되므로 구현 방법이 복잡하고, 유지보수가 어렵다는 문제점을 가지고 있다. 또한 중·소규모의 네트워크보다는 대규모 분산네트워크에 적합한 방법이다[6]. ASP 사업에서 데이터베이스 연동에 상기 코바를 이용하는 경우 시스템에 부하만 부과될 뿐더러 ASP 고객은 ASP 사업자에게 자신의 데이터베이스 서버의 네트워크 경로와 스키마를 노출해야 되는 보안상의 문제가 있다. 이에 본 연구에서는 이 부분을 회피 할 수 있도록 가볍고 보안이 가능한 응용프로그램을 이용한다.

2.2 XML을 이용한 데이터 전송

HTTP 통신 프로토콜을 이용하는 웹 상에서 데이터전송과 데이터교환은 CORBA를 사용하는 것 보다 XML이 효과적으로 사용되고 있다[7]. 또한 XML은 특정 도메인에 해당되는 태그를 정의할 수 있어 메타 데이터 관리가 용이하다. 이러한 XML을 이용한 데이터베이스 연동의 연구로 권은정과 용환승[9]은 관계형 데이터베이스의 호스트 이름, 특정 DBMS의 버전 등의 메타 데이터를 XML 문서로 생성한 후 XML 전용 객체지향 데이터베이스에 저장하고, 메타 데이터를 이용하여 세부정보를 검색하는 연구를 수행하였다. 권은정 외[10]는 표준이 정해지지 않은 OLAP(On Line Analytical Processing) 제품들간의 공통으로 사용할 수 있는 메타 데이터 교환 모델로 XML을 사용하는 연구를 XML 전용 데이터베이스를 이용하여 수행했다. XML 문서 저장에 관한 연구로 정상혁 외[11]은 XML 문서 저장 시스템을 기반으로 기업간의 거래에 사용된 XML 문서를 XML 전용 데이터베이스에 저장하는 B2B 통합 시스템을 구축했다. XML을 이용한 상기 연구들은 XML 전용 데이터베이스를 사용한다. 그렇지만 ASP 사업의 경우는 데이터베이스도 상용 관계형 데이터베이스를 사용할 뿐만 아니라, 데이터를 수신하는 ASP 데이터베이스에서 다수의 ASP 고객의 메타 데이터를 관리해야 하는 회원 관리의 부가적인 문제가 발생한다. 또한, 고객은 메타 데이터 내에 보안상 노출 시키고 싶지 않은 정보가 존재 할 경

우 ASP 응용프로그램의 이용 자체를 기피 할 수 있다. 이상의 이유 등으로 ASP 사업의 데이터베이스 연동의 경우는 상기 방법과는 다른 데이터베이스 연동 아키텍처(architecture)의 연구가 필요하다.

2.3 ASP 응용프로그램

ASP 사업은 국내와 해외에서 삼성 SDS[13], 코인텍[14], 비즈메카[15], USi[16] 등의 업체에서 엔터프라이즈(enterprise), 협업(collaborative), 퍼스널(personal) ASP 형태로 분류하여 제공되고 있다.

엔터프라이즈 ASP 형태에서는 ERP, SCM, CRM 등, 협업 ASP 형태는 그룹웨어(Groupware), 전자우편, 화상회의 등, 퍼스널 ASP 형태는 사무용 프로그램이나 게임 등과 같은 종류의 응용프로그램을 제공하고 있다. 상기 언급된 다양한 ASP 응용프로그램의 종류에서 데이터베이스를 필요로 하고, 특히 ERP 응용프로그램 공급의 경우 인사, 급여, 재고, 생산 관리 등의 데이터를 ASP 고객의 데이터베이스로부터 전송 받아야 한다. 또한, ASP 고객이 분석 CRM을 위해 ASP 응용프로그램의 고객분석 기능을 이용하고자 하는 경우, CRM 응용프로그램은 고객 정보와 고객 거래 정보 등을 필요로 하므로 ASP 고객은 이 데이터를 전송해야 한다. 이와 같은 사례가 본 연구의 필요성을 입증해 준다.

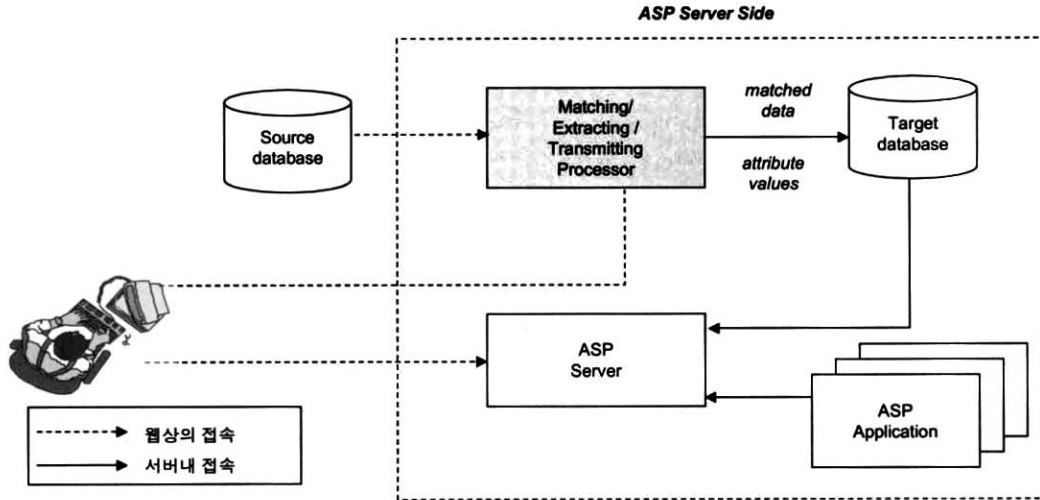
3. 시스템의 기능 및 설계

먼저 제안하는 시스템의 설계를 서술하기에 앞서, 데이터베이스의 연동에 관한 핵심 기능이 ASP 측에서 수행되는 경우의 문제점을 3.1에서 검토하고, 이러한 문제점을 극복하며 보다 개선된 연동을 위한 필요 기능을 3.2에서 정의하고, 3.3은 이 기능을 만족 시키는 시스템의 아키텍처를 설계한다.

3.1 ASP 웹 사이트에서의 데이터 전송

본 연구에서 데이터베이스 간의 연동 주체로서 두 개의 데이터베이스를 대상으로 한다. 이때 고객의 데이터베이스를 '소스 데이터베이스(source database)'라 부르고, ASP 측의 데이터베이스를 '타겟 데이터베이스(target database)'로 부를 것이다. 그리고 데이터베이스의 연동이란 ASP 응용프로그램이 운영하는 데이터베이스로 ASP 회원 자신의 데이터베이스의 일부 자료를 제공하는 것을 의미한다.

ASP 웹 사이트에서 데이터를 전송하는 방법은 두 데이터베이스 간에 온라인(on-line) 상태로 데이터를 전송을 하는 경우이다. 이를 위한 프로세서는 (그림 2)와 같이 ASP 웹 사이트에 위치하게 된다. 고객은 이 프로세서의 안내에 따라 데이터 전송을 한다.



(그림 2) 속성 매칭 및 데이터 전송 프로세서가 ASP 웹 사이트에 위치하는 경우의 시스템 구성도

전송을 위해 해야 할 일은 세 가지로 나눌 수 있다. 첫 번째로 서로 다른 속성명을 매칭 시키는 것이고, 두 번째로 속성명 매칭 후에 정보 추출을 하고, 마지막으로 추출된 정보를 타겟 데이터베이스에 전송하는 것이다. 아울러 매칭된 데이터베이스명, 테이블명, 속성명에 대한 정보는 최초 속성 매칭 이후 새로운 데이터를 전송할 때, 속성 매칭을 반복하지 않고 사용할 수 있도록 하기 위해 타겟 데이터베이스에 저장한다.

<표 1>은 매칭, 추출, 전송 그리고 저장 기능을 갖는 프로세서가 ASP 웹 사이트에 위치할 경우의 데이터 처리 절차이다. STEP 2에서 보는 바와 같이 고객은 자신의 데이터베이스를 로그인 할 때 데이터베이스 서버 IP 주소, 데이터베이스 로그인 정보, 데이터베이스명, 테이블명 등을 온라인상에서 입력된다. 이 정보는 네트워크 상으로는 보안 유지를 시킬 수 있겠으나, ASP 측에 노출될 수밖에 없다. 이는 ASP 서비스 활성화를 저해할 수 있다.

<표 1> ASP 웹 사이트에서의 데이터 연동 절차

- STEP 1 : 타겟 데이터베이스가 있는 ASP 웹사이트 접속
- STEP 2 : 클라이언트의 소스 데이터베이스 로그인
- STEP 3 : 타겟 데이터베이스에서 요구하는 속성명으로 클라이언트 소스 데이터베이스 속성명 검색 및 등록
- STEP 4 : 매칭된 속성명의 속성치 추출 및 전송
- STEP 5 : 전송 자료 및 매칭관련 자료(데이터베이스명, 테이블명, 속성명) 저장
- STEP 6 : 소스 데이터베이스 접속 종료
- STEP 7 : ASP 응용프로그램의 사용
- STEP 8 : ASP 사이트 접속 로그아웃

3.2 시스템 요구 기능

본 연구에서는 3.1에서 도출된 문제점을 극복 하며 보다 개선된 연동을 위한 시스템이 갖추어야 할 기능은 다음과

같다.

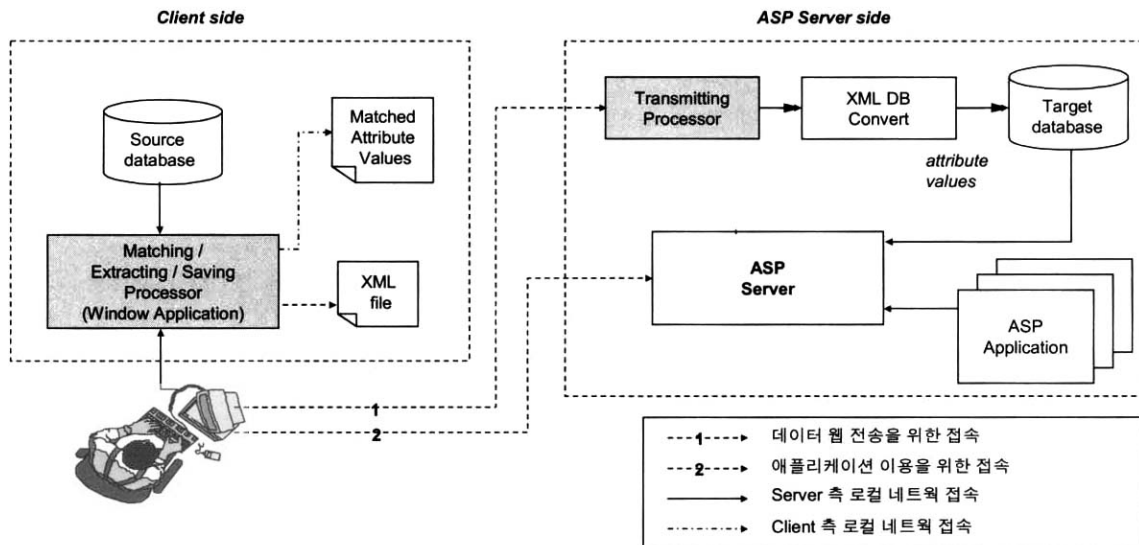
- 데이터베이스의 연동 시에 전송되는 데이터들은 속성치가 동일하나 속성명칭이 다른 경우의 연동을 지원해야 한다. 이러한 속성명칭 불일치의 문제는 데이터베이스 설계자가 데이터베이스를 설계할 때 서로 다른 속성명칭 사용하기 때문에 발생된다. 서로 다른 데이터베이스의 연동을 지원하기 위해서는 동일한 속성치를 갖는 속성명칭 간의 속성 매칭을 수행한다. 속성 매칭은 데이터베이스의 수가 작으면 비교적 쉽게 처리할 수 있으나 데이터베이스 수가 증가하면 속성 매칭이 거의 불가능하다는 문제점이 야기된다.
- 고객이나 ASP의 데이터베이스 연동 시스템이 CORBA 등과 같은 미들웨어를 사용하지 않고 연동 시스템 구축을 가볍게 한다.
- ASP 사업의 활성화를 위하여 데이터 연동과 데이터 전송에 따른 고객의 데이터베이스 관련 보안 정보를 ASP에게 노출 시키지 않는다.
- 데이터 연동 관련 메타 데이터의 관리를 고객 측에서 관리하여, ASP 서버의 고객 관리 부하를 낮춘다.
- ASP 고객측에서 사용할 클라이언트 프로그램은 SQL을 모르는 사람도 사용이 가능하도록 하고, 이 클라이언트 프로그램의 제공은 ASP 웹사이트를 이용하여 배포가 용이하도록 한다.
- 속성의 타입이 다른 경우, 빈 데이터가 존재하는 경우, 제어문자 및 유니코드 문자에 대한 처리 문제, 그리고 데이터 변환 과정에 계산이 필요한 경우 등의 다양한 문제를 처리 할 수 있어야 한다

위에서 제시한 기능을 포함하는 데이터 전송 방법은 다음 절에서 기술한다.

3.3 보안을 고려한 데이터 전송

3.1에서 소개한 방법의 문제점은 소스 데이터베이스의 접속을 위한 로그인 정보가 ASP 사업자 에게 웹상에서 노출될 위험이 있어 ASP 회원 확보에 어려움이 될 수 있다. 웹상의 보안 문제는 이를 위한 새로운 보안 시스템을 도입할 수 있으나, 본 연구에서는 가급적 다른 보안 시스템을 이용

하지 않고 ASP 고객이 용이하게 사용할 수 있는 저 비용의 시스템을 제안하려고 한다. 이 방법의 중심적인 접근법은 고객 데이터베이스로부터의 데이터 매칭 및 추출 프로세스를 오프라인(off-line)화 하고, 고객 사이트에서 수행하게 하는 것이다. 데이터가 추출되면, 이후 ASP 웹 사이트에 접속하여 추출된 데이터를 (그림 3)과 같이 전송한다.



(그림 3) 속성 매칭 및 추출 프로세서가 ASP 고객측에 위치할 경우의 구성도

<표 2> 매칭 및 전송 프로세서가 클라이언트 응용프로그램으로 존재하는 경우 데이터 연동 절차

- STEP 1: ASP 웹 사이트와 오프라인 상태에서 클라이언트용 윈도우 응용프로그램으로 고객 자신의 소스 데이터베이스 접속
- STEP 2: ASP 타겟 데이터베이스에서 요구하는 속성명에 해당되는 소스 데이터베이스 및 테이블 검색
- STEP 3: 매칭 정보를 이용 XML 파일로 데이터 추출 및 저장과 속성 매칭에 관련된 메타 데이터 자동 저장
- STEP 4: ASP 사이트 접속
- STEP 5: 생성된 XML파일 ASP 사이트로 업로드 및 타겟 데이터베이스 저장
- STEP 6: ASP 응용프로그램 사용
- STEP 7: ASP 사이트 접속 로그아웃

속성명 매칭은 네 단계로 구성되고 ASP 사업자가 요구하는 속성치와 매칭하는 속성명의 정보 검색을 고객의 플랫폼상에서 처리하도록 한다. 첫 번째 단계는 ASP 웹 사이트와 오프라인 상태에서 ASP 웹 사이트로부터 다운로드된 클라이언트 응용프로그램을 사용하여 고객 자신의 소스 데이터베이스를 네트워크 접속한다. 두 번째 단계는 클라이언트 응용프로그램에 제시된 타겟 데이터베이스가 요구하는 속성명을 참고하여 고객이 자신의 관련 데이터베이스 및 테이블을 검색하고, 그 속성명을 상기 언급된 클라이언트 응용프로그램의 타겟 데이터베이스의 속성명에 매칭시킨다.

세 번째 단계는 매칭된 결과를 이용하여 데이터를 추출하고 XML 파일로 저장한다. 그리고 매칭된 정보는 매칭 정보 수정이나 새로운 데이터 추출 시 속성 매칭 과정을 반복하지 않도록 고객의 플랫폼에 메타 데이터 파일로 자동 저장한다. 마지막으로 저장된 XML 파일을 ASP 웹사이트에 접속한 후 업로딩(uploading)하여 타겟 데이터베이스에 저장한다. 이러한 업로딩 과정은 고객의 매칭 데이터 값이 변화가 있을 경우에는 항상 필요하다. 즉, 고객 데이터의 변화가 없을 시는 오프라인 과정 및 파일 업로딩 과정 없이 곧바로 ASP 응용프로그램을 사용하게 된다. <표 2>는 상기 언급된 과정을 단계별로 설명한다.

3.4 기존 방법과의 비교

현재 ASP 고객의 데이터를 전송 받아야 하는 ASP 사업에서, ASP 고객의 데이터 전송은 ASP 고객이 엑셀 파일 형태의 데이터 파일을 ASP 사업자에게 전달한다. ASP 사업자는 전달된 엑셀 데이터를 ASP 데이터베이스에 저장한다. 이는 ASP 고객에게 추가적인 비용과 ASP 응용프로그램 사용의 지연을 초래하고 있다. 그래서 본 연구에서는 이러한 전송 방법이 아닌, ASP 고객이 자신의 데이터베이스에서 XML 파일로 추출된 데이터를 파일 ASP사이트에 업로딩 한 후, ASP 사업자와는 무관하게 ASP 데이터베이스

에 저장하는 방법을 이용한다. ASP 고객이 엑셀 파일을 이용하는 경우와 본 연구가 제안하는 방법에 대하여 데이터 추출, 데이터 전달, ASP 서버상의 데이터베이스로 데이터 저장의 3단계를 비교 설명하면 다음과 같다.

3.4.1 데이터 추출

- 엑셀 파일로 추출하는 경우
 - ASP 고객은 ASP 애플리케이션에서 요구하는 데이터를 ASP 사업자가 제공한 자료를 통해서 확인한 후에 자신의 데이터베이스에 접속하여 DTS를 이용하여 데이터를 추출하거나 직접 SQL 쿼리문을 작성하여 추출한다. 추출 결과는 엑셀 파일로 저장하는 export처리를 해야 한다.
- 본 연구 제안 방법
 - ASP 고객이 클라이언트 응용프로그램의 우측 화면에 출력된 ASP의 데이터베이스 요구하는 속성들의 종류를 이해하고, 클라이언트 응용프로그램의 우측 화면에 트리 구조로 출력된 고객 자신의 데이터베이스를 검색한다. 검색 중 매칭되는 속성이 발견되면 데이터베이스 속성명을 드래그앤드랍(drag & drop)으로 속성 매칭 입력란에 매칭 시킨다. ASP 고객은 ASP 데이터베이스가 요구하는 속성에 대한 속성 매칭 완료 후 자신의 데이터베이스에서 데이터를 추출한다.
- 각각의 비교
 - 엑셀을 이용하는 방법에서 여러 테이블에 분산된 경우에 ASP 고객은 MS-SQL Server 엔터프라이즈 관리자의 DTS 사용법과 SQL 쿼리문 문법을 숙지한 전문가의 도움을 필히 받아야 한다. 본 연구가 제안하는 방법에서는 ASP 고객은 SQL문 작성없이 다운로드 프로그램을 실행하여 나타난 유저 인터페이스상에서 간단히 드래그 & 드랍 기능만으로 데이터를 추출하고 확인 버튼으로 XML 파일로 저장 할 수 있다. 왜냐하면 다운로드 프로그램을 통하여 접속한 고객의 임의의 데이터베이스 인스턴스의 데이터베이스, 테이블 및 속성 정보들은 유저 인터페이스상에 모두 트리(tree) 형태로 그 구조정보가 출력되고 사용자는 이 트리를 단순한 클릭 행위로 항해하면서 필요한 속성을 발견한다. 속성이 발견되면 마우스로 이 속성명을 클릭 선택하여 유저 인터페이스 상에 이미 출력 제시된 ASP 애플리케이션 및 타겟 데이터베이스가 요구하는 속성명 옆의 준비된 빈 공간 자리까지 끌어 놓으면(drag& drop) 매칭 행위는 종료된다.
 - 모든 필요 속성의 매칭 행위가 종료되면 고객 데이터베이스로부터의 데이터 추출과 저장은 화면 상의 종료 버튼을 클릭함으로써 다운로드 프로그램이 직접 추출을 수행한다.

- 따라서 상기 2가지 추출 작업은 모두 수작업 처리되지만 기존의 엑셀을 이용하는 과정의 수작업 절차는 제안된 방법에 비하여 상대적으로 번거로운 작업일 뿐 아니라 추출과정에서 DTS나 SQL 등을 처리할 수 있는 DB 운영자가 반드시 필요하다. 반면에 후자의 제안된 방법은 드래그앤드랍 방식을 통하여 마우스 조작만 하면 되고 데이터베이스 운영자가 없어도 가능하다.

3.4.2 데이터 전달

- 엑셀 파일의 전달
 - 데이터가 저장한 엑셀 파일은 ASP 사업자에게 직접 전달되거나 인터넷 서비스(e-mail, ftp 등)로도 전송 가능하다.
- 본 연구의 XML 파일 전달
 - 고객에 의하여 추출된 데이터를 내장한 XML 문서가 저장된 컴퓨터로부터 직접 ASP 사이트로 웹상에서 업로딩 된다.

3.4.3 ASP 서버상의 데이터베이스로 데이터 저장

- 엑셀 파일을 이용하는 방법
 - ASP 사업자는 전달 받은 엑셀 파일의 바이러스 유무, 데이터의 유효성을 검토 확인한 후에 ASP 서버측의 데이터베이스의 DTS를 이용하거나 SQL문을 작성하여 파일내의 데이터를 데이터베이스로 저장시키는 수작업을 행해야 한다.
- 본 연구 제안 방법
 - ASP 고객에 의하여 업로드된 XML 파일은 ASP 데이터베이스에 자동 저장된다.
- 각각의 비교
 - 엑셀 파일을 이용하는 방법에서 고객의 수, 데이터베이스를 사용하는 ASP 애플리케이션의 수, 동일한 고객의 데이터 업그레이드 등의 경우가 증가할 경우에 ASP 사업자의 파일 정보의 적합성 판독 및 데이터베이스로의 수작업 처리업무는 과부하가 될 수 있다. 이는 서비스의 소홀이나 지연으로 연결되어 ASP 사업의 활성화에 저해 요인이 된다. 본 연구팀의 조사에 의하면 데이터 저장을 위해 약 2주 정도의 지연이 흔히 발생할 수 있음을 국내 사업자에게로부터 확인한 바 있다.

4. 시스템 구현

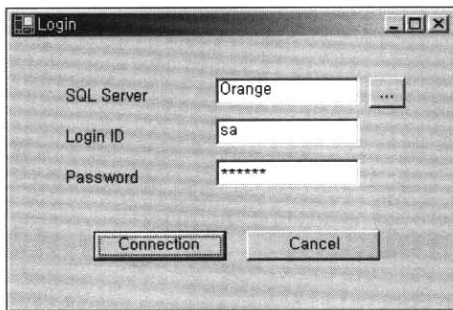
본 연구에서는 오프라인 상에서 소스 데이터베이스 속성명 검색 및 ASP 측의 데이터베이스의 속성과의 매칭 그리고 XML 문서로 고객의 데이터를 추출하여 ASP로 전송하는 기능을 갖는 D2D 프로토타입 시스템을 구현하였다. 다음은 시스템의 구현된 기능에 대한 두 데이터베이스의 연동

을 위한 시스템의 구동 단계별 사례이다. 먼저 고객의 플랫폼 상에서 이용되는 ASP 서버로부터 다운 받아 ASP 서버와 오프라인 상태에서 고객이 사용하는 클라이언트 프로그램을 4.1, 4.2 그리고 4.3항에서 언급하고, 4.4에서는 ASP 서버측의 전송된 데이터의 포스트 프로세싱(post-processing) 과정을 언급한다.

4.1 고객 소스 데이터베이스 로그인

고객이 속성 매칭을 위해 클라이언트 프로그램을 이용하는 첫 번째 단계로 자신의 소스 데이터베이스에 접속을 필요로 한다. 소스 데이터베이스 접속은 고객의 플랫폼에 ASP 서버로부터 다운로드 된 클라이언트 프로그램을 이용한다. (그림 4)는 이 클라이언트의 로그인 화면을 보여준다. 이 화면을 통하여 소스 데이터베이스 연결을 위한 데이터베이스 인스턴스 명이나 IP주소, 데이터베이스에 접근권한이 있는 아이디(ID)와 암호가 입력된다. 이들 값의 입력은 먼저 데이터베이스 서버의 선택으로 시작된다.

이는 (그림 4)의 GUI 화면 버튼을 클릭했을 때에 고객의 플랫폼의 로컬 네트워크 상에 인식 가능한 모든 데이터베이스 인스턴스 리스트 중에서 선택한다. 데이터베이스 인스턴스(instance) 객체를 선택한다. 고객이 소스 데이터베이스의 인스턴스 명이나 IP 주소를 알고 있는 경우는 인스턴스 명을 검색하지 않고 직접 화면 상에서 입력할 수 있다.



(그림 4) 사용자 로그인 GUI 화면

4.2 속성 매칭

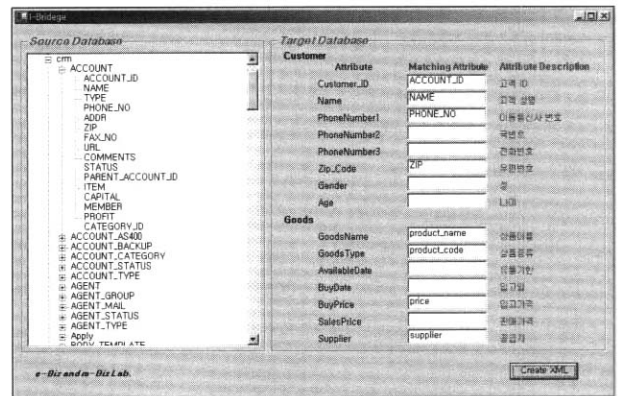
속성명 매칭을 위한 클라이언트 응용프로그램의 메인화면은 (그림 5)와 같다. (그림 5)의 우측은 타겟 데이터베이스에서 필요로 하는 속성명, 매칭 시킬 소스 데이터베이스의 속성명 입력란과 속성에 대한 설명으로 보여준다. 화면 좌측은 접속된 소스 데이터베이스를 데이터베이스명, 테이블명, 속성명 순서의 트리구조 형태로 출력한 결과이다.

고객은 우측 화면으로부터 ASP 웹 사이트의 타겟 데이터베이스가 요구하는 속성들의 종류를 이해하고, 우측의 요구 속성명에 매칭되는 속성명을 좌측에서 탐색한다. 탐색으로 발견된 속성명은 이를 우측 매칭 속성 입력란으로 각각 드래그 앤 드롭(drag and drop)하여 입력시킴으로써 매칭이 이

루어진다.

(그림 5)의 속성 매칭 입력란에 처리된 결과는 타겟 데이터베이스가 요구하는 Customer_ID와 Name 등의 속성명을 소스 데이터베이스에서 검색하여, Customer_ID는 ACCOUNT_ID로 Name은 NAME으로 매칭된 예이다.

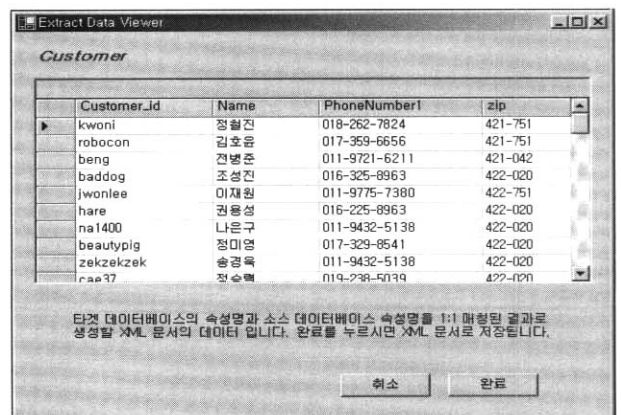
속성 매칭된 정보는 속성치 추출이 종료될 때 메타 데이터 파일로 저장된다. 이를 이용하게 되면 ASP 고객이 과거에 클라이언트 응용프로그램을 이용한 속성 매칭을 수행한 경우, 클라이언트 응용프로그램 실행 시 고객의 플랫폼에 저장된 메타 데이터를 읽어 속성 매칭 결과를 화면상에 보여준다. 속성치 추출이 필요할 때마다 속성 매칭을 반복적으로 수행하지 않고 속성치를 추출할 수 있게 한다.



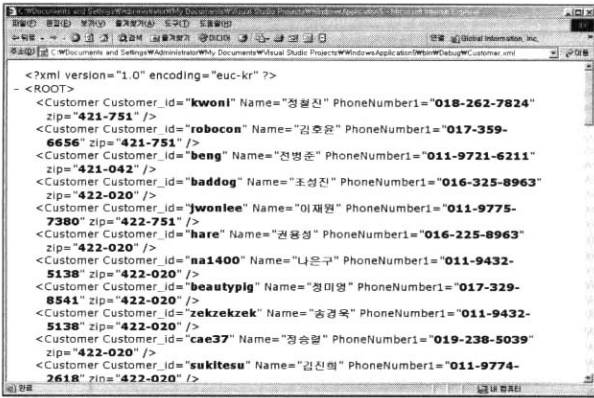
(그림 5) 속성 매칭 및 데이터추출 위한 D2D 클라이언트 프로그램

4.3 속성치 추출 및 XML 문서 저장

모든 속성명의 일대일 매칭 과정이 전단계에서 완료되면, 관련 속성명들의 속성치는 일괄적으로 추출되어 XML 파일로 저장한다. 저장된 XML 파일은 XML 문서 구조에 타당한(valid) 형식을 갖는다. 이때에 타겟 데이터베이스의 요구 속성명에 일대일 매칭된 소스 데이터베이스의 속성명은 고객 플랫폼 상에 메타 데이터로 저장된다.



(그림 6) 추출 속성치 확인 및 XML 생성



(그림 7) 속성 매칭으로 생성된 XML문서 데이터

(그림 6)은 두 데이터베이스의 속성명 매칭 후 ASP 고객의 데이터베이스인 소스데이터베이스로부터 추출된 속성치 확인 단계이다. 이때 잘못된 데이터가 추출되었다면 속성 매칭 단계로 이동 속성 매칭을 재 수행한다. ASP 고객은 추출된 속성치 확인 단계를 거친 후, 추출 속성치를 XML 문서로 저장한다. 이때에 속성 매칭 결과의 정보는 메타 데이터로 저장된다. (그림 7)은 속성 매칭 결과로 생성된 XML 문서의 데이터이다. 속성치 추출, XML 문서 저장 그리고 메타 데이터 저장은 백엔드(back-end) 인터페이스 방식으로 실행된다.

4.4 추출 속성치의 ASP 전송

XML 파일로 추출된 속성치 데이터는 ASP로 전송 되어야 한다. 이를 위해 ASP 고객은 ASP 웹 사이트에 접속한 후, ASP 웹 사이트의 전송 프로세서를 이용하여 XML 파일을 전송한다. ASP 서버에 전송된 데이터는 XML DB 변환기에 의하여 타겟 데이터베이스로 입력 저장되는데 저장 전에 읽혀진 XML 문서를 고객의 웹 브라우저 화면에 출력하여 확인 후 저장한다.

데이터 저장과정에서 데이터 속성 타입 불일치 문제, 빈 데이터가 존재하는 경우, 제어문자 및 유니코드 문자에 대한 처리 문제, 그리고 데이터 변환 과정에 계산이 필요한 경우 등의 다양한 문제가 발생한다. 첫째, 데이터 속성 불일치 문제의 해결 방법은 타겟 데이터베이스의 데이터 형식을 varchar 형식으로 설계한다. 둘째, 제어문자 및 유니코드 문자에 대한 처리 문제는 ASP 데이터베이스 시스템에서 non-ASCII 문자를 저장할 수 있도록 환경 설정한다. 셋째, 빈 데이터가 존재하는 경우는 빈 데이터를 NULL값으로 처리해서 ASP 데이터베이스에 저장한다. 마지막으로 데이터 변환과정에 계산이 필요로 하는 경우는 클라이언트 프로그램에서 데이터 추출시 ASP 데이터베이스가 요구하는 형태로 변경하여 추출하므로 저장과정에서는 변환을 필요로 하지 않는다.

4.5 구현 환경

다음은 본 논문에서 시스템 구현에 사용된 개발 환경을 살펴보면 소스 데이터베이스와 타겟 데이터베이스는 MSSQL Server 2000을 사용하였다. 그리고 클라이언트 응용프로그램 개발도구로는 .Net C#을 ASP 애플리케이션은 ASP.Net과 C#을 이용하였다.

5. 결 론

본 논문은 ASP 사업의 제공되는 특정 응용프로그램이 데이터베이스를 이용하며, 이 데이터베이스가 ASP 고객이 보유하고 있는 데이터베이스의 일부 데이터를 필요로 하는 경우에 관한 데이터베이스 연동 연구로서, 연동 방법론 제안과 이를 검증하기 위한 프로토타입 시스템 D2D의 구현을 기술하였다. 논문에서 다룬 내용은, 첫째, 연동 시에 두 데이터베이스에서 공유해야 하는 데이터에 대하여 각 데이터베이스의 속성명이 상이한 경우에 대한 속성 매칭 과 데이터 추출에 대한 방법론이고, 둘째는 네트워크상에서 데이터베이스 연동에 필요한 고객의 데이터베이스의 경로 정보가 ASP 사업자에게 노출 되지 않도록 하는 것이다.

본 연구에서 제시한 고객의 데이터베이스 데이터가 ASP 사업자에게 전송되는 과정은 크게 세 가지로 구성된다. 첫째는 ASP 사이트에서 다운로드 받은 클라이언트 프로그램을 이용하여 오프라인 상태로 고객 데이터베이스로부터 추출, 둘째는 이 파일의 웹을 이용한 ASP 사업자 사이트로의 업로딩, 그리고 나머지는 업로딩된 데이터의 ASP 사업자의 데이터베이스로의 자동 저장이다.

이를 통한 기존의 엑셀 등과 같은 파일을 이용하여 ASP 사업자에게 전달 되는 과정에 비교 했을 때 장점은 다음과 같다. 먼저 고객 데이터베이스로부터의 데이터 추출 과정에서의 장점은 데이터베이스 운영자의 도움 없이 다운로드 프로그램의 유저 인터페이스를 이용하여 접속된 데이터베이스의 속성 정보를 용이하게 추출할 수 있다는 것이다. 이는 다운로드 프로그램의 유저 인터페이스에 이미 ASP 사업자가 요구하는 속성정보가 제시되어 있어 이에 매칭 가능한 속성명을 접속한 고객의 데이터베이스로부터 추출된 유저 인터페이스에 제시되는 테이블 및 속성 정보의 트리 구조를 마우스로 쉽게 항해하면서 드래그앤드랍 방식으로 매칭이 가능하다. 한번 매칭된 속성들은 메타 데이터로 자동 관리되어서, 차후 데이터 추출을 반복시에는 자동 이용된다. 이 때는 단순히 데이터 추출 과정만 반복된다. 두 번째의 추출된 파일의 데이터 전송 단계도 엑셀의 경우는 e-mail 등을 이용하나, 본 시스템에서는 파일 업로딩 과정으로 그 난이도에서는 차이가 별로 없다. 세 번째 과정은 ASP 사업자의 데이터베이스로 이 파일 정보가 삽입되는 경우이다. 이 경우 본 제안된 방법은 포스트 프로세서가 자동처리 하

므로 ASP 사업자가 엑셀 파일 처리하기 위하여 데이터베이스 운영자의 도움을 받아야 하는 경우가 생략된다.

- 데이터 전송이 수작업으로 ASP 측에 입력되지 않고 시스템 수준에서 처리되어 입력의 과오와 시간이 줄어든다.
- ASP 사업자나 고객은 데이터베이스 운영자의 도움을 직접 받지 않고도 고객의 데이터를 용이하게 사업자의 데이터베이스로 전송할 수 있다. 특히 ASP 사업자는 데이터 저장단계에서는 부과되는 업무가 거의 없다.
- 고객이 사용하는 프로그램은 ASP 사업자로부터 다운로드 되므로 배포가 용이하다.
- ASP 사업의 활성화를 위하여 데이터 연동과 데이터 전송에 따른 고객의 데이터베이스 관련 보안 정보를 ASP에게 노출시키지 않는다.

비록 상기 장점이 있으나 ASP 사업자는 다운로드용 클라이언트 프로그램의 개발과 ASP 서버 사이트에서 자신의 데이터베이스로 고객파일의 자동저장을 위한 포스트 프로세서의 개발 부담이 있다. 그러나 이 부담은 본 논문에서 기술된 기존의 엑셀을 이용하는 경우의 사업자의 부담과 비교해 보면 경미하다 할 수 있다. 한 고객이 자신의 데이터를 자주 갱신하는 횟수, 고객의 수에 따라서 사업자의 데이터베이스 작업은 비례하여 증가되며 또한 이러한 데이터베이스를 사용하는 애플리케이션의 수가 증가하는 경우까지 고려하면 사업자의 업무인 고객의 데이터 파일의 수작업을 위한 파일 바이러스 검증, 데이터의 유효성 검증과 데이터 저장을 위한 데이터베이스 운영 수작업 업무는 결코 과소 평가 할 수 없을 것이다.

이러한 다운로드하는 클라이언트 프로그램의 사용은 두 데이터베이스가 온라인 상에서 데이터 전송이 이루어지는 것에 비하면 불편한 방법이다. 하지만 현재 웹상에서 자사의 시스템이 아닌 고객의 데이터베이스를 사업자의 데이터베이스와 연동을 하기 위해서는 고객이 웹상에서 사업자가 준비한 웹 페이지를 이용하여 연동을 해야 하며, 이 경우에 고객은 자신의 웹상의 데이터베이스 경로를 입력해야 하는데 이는 사업자에게 노출이 된다. 이러한 보안상의 부담을 안게 되므로 결국 본 연구에서 제시된 클라이언트 프로그램을 이용한 오프라인으로 고객 자신의 데이터베이스 접속은 웹상의 데이터베이스 경로 정보의 노출을 원천적으로 봉쇄할 수 있는 보안상의 유리함도 누리게 된다. 물론 이때 말하는 보안은 데이터 전송시의 기밀성이나 무결성에 관한 부분과는 다른 것이다. 단지 두 데이터베이스 연동시의 데이터베이스 경로 기밀에 대한 것에 제한된다. 실제 본 연구된 시스템의 데이터 업로딩의 실질적인 서비스에서는 XML 문서의 보안 기능 등을 이용하여 전송되는 데이터의 암호화가 필요할 것이다.

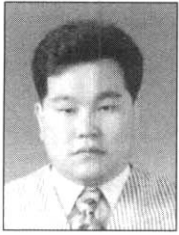
현재 데이터베이스를 필요로 하는 CRM, ERP 등의 ASP

애플리케이션의 수가 증가되는 추세에 있는바 제시된 본 연구의 데이터베이스 연동에 관한 프레임워크는 ASP 비즈니스에 하나의 활성화 방향을 제시할 것으로 기대한다.

향후 연구로는 다양한 데이터베이스별 연동과 실 사례의 적용이 필요하다.

참 고 문 헌

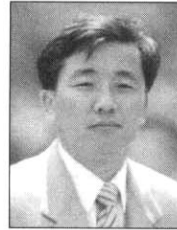
- [1] Thomas Wiedemann, "Web-based simulation : Simulation application service providing (SIM-ASP)," Proceedings of the 33nd conference on Winter simulation, pp.623-628, Dec., 2001.
- [2] Amit P. Sheth, James A. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," ACM Computing Surveys (CSUR), Vol.22, No.3, 1990.
- [3] Ramon Lawrence, Ken Barker, "Integrating relational database schemas using a standardized dictionary," Proceedings of the 2001 ACM symposium on Applied computing, 2001.
- [4] Erhard Rahm, Philip A. Bernstein, "A survey of approaches to automatic schema matching," The VLDB Journal The International Journal on Very Large Data Bases, Vol.10 No.4, pp.334-350, 2001.
- [5] Mark Roantree, John Murphy, Wilhelm Hasselbring, "The OASIS multidatabase prototype," ACM SIGMOD Record, Vol.28, No.1, 1999.
- [6] Robert Orfali, Dan Harkey, Jeri Edwards, The essential distributed objects survival guide 2nd edition, Prentice Hall, 1996.
- [7] <http://www.omg.org/news/whitepapers/whtsonwp.htm>.
- [8] 변광준, "EasyConnection : 이질 데이터베이스들의 효율적 연동방식", 한국전자거래(CALS EC)학회, 한국전자거래(CALS/EC)학회지, Vol.4, No.3, pp.235-246, 1999.
- [9] 권은정, 용환승, "데이터베이스 : XML 을 기반으로 한 관계형 데이터베이스 메타데이터 리파지토리 설계 및 구현", 정보처리학회논문지D, Vol.9-D, No.1, pp.1-10, 2002.
- [10] 이인기, 이민수, 용환승, "웹 기반의 OLAP 메타데이터 교환 시스템의 설계 및 구현", 정보처리학회논문지D, Vol.9-D, No.6, pp.971-980, 2002.
- [11] 정상혁, 주경수, 이정수, "멀티미디어 데이터베이스 : XML 데이터베이스 시스템을 기반으로 한 B2B 통합 시스템 개발", 멀티미디어학회논문지, Vol.6, No.1, pp.1-14, 2003.
- [12] 이순철, 김성훈, "정보시스템관리(1) : ASP의 효과적 도입 방안", 2002 한국경영정보학회 추계학술대회, Vol.2002, No.0, pp.177-188, 2002.
- [13] <http://www.sds.samsung.co.kr/product/unierp/index.php>.
- [14] <http://www.kointech.com/>.
- [15] <http://www.bizmeko.co.kr/>.
- [16] <http://www.usi.com/>.



김 호 윤

e-mail : robocon@shinbiro.com
1996년 조선대학교 기계공학과(학사)
1998년 조선대학교 대학원 기계공학과
(공학석사)
1999년~현재 인하대학교 대학원 첨단정밀
공학과 박사과정

관심분야 : XML, 무선 인터넷, e비즈니스



이 재 원

e-mail : jwonlee@inha.ac.kr
1977년 서울대 기계공학과(학사)
1979년 KAIST 생산공학 전공(공학석사)
1984년 ECN(Ecole Centrale de Nantes,
France) 공대 자동화공학(공학박사)
1990년~1991년 Carnegie Mellon
University 방문연구

1985년~현재 인하대학교 산업공학 전공 교수

관심분야 : e-비즈니스에서 모바일 CRM, 모바일 광고 및 유비
쿼터스 비즈니스