

기업 정보 시스템간 효율적인 데이터 공유를 위한 통합 도구 개발

한관희*(경상대학교 산업시스템공학부), 박찬우(경상대학교 기계항공공학부/BK사업단),
최운집(수성기체(주)), 이상한((주)CIES)

The Development of an Integration Tool for the Data Sharing Among the
Enterprise Information Systems

K. H. Han(Indus. Eng. Dept., GSNU), C. W. Park(Aero. Eng. Dept., GSNU), U. G. Choi (SooSung Co.), S. H. Lee (CIES Co.)

ABSTRACT

Recently, many enterprises are introducing EAI (Enterprise Application Integration) technologies for integrating heterogeneous enterprise information systems. Among EAI levels, data-level integration is relatively straightforward and most popular. However, current commercial solutions have complex functionalities and are expensive for implementing the data integration tasks. Also, they have their own proprietary architectures and have a restricted interoperability. Proposed in this paper is the development of data integration middleware for facilitating data exchanges between the heterogeneous information systems. The main feature of this middleware is a explicit mapping of meta data about the relationships between source and target data. Based on this mapping, users who do not have expertise in information technology at the small & medium enterprises can easily handle data exchange tasks between information systems.

Key Words : EAI (기업응용시스템 통합), data integration (데이터통합), data replication (데이터 복제), integration middleware(통합 미들웨어)

1. 서론

기업 환경의 급속한 변화로 인해 기업간 경쟁이 심화되면서 이제 정보 시스템은 기업 경영을 단순히 지원하는 수준에서 벗어나 기업 생존의 전략적 도구로 자리매김하고 있다. 1990년대부터 기업 관리 영역을 전체적으로 연결하는 ERP (Enterprise Resource Planning) 시스템의 도입이 추진되어 대다수의 기업이 기업 전략에 적극적으로 활용하고 있으며 최근에는 설계 부문의 PDM (Product Data Management) 시스템, 생산 현장의 MES (Manufacturing Execution System), 마케팅 부문의 CRM (Customer Relationship Management) 시스템 등의 도입도 활발해지고 있다.

이와 같이 기업의 전략적 도구로서의 정보 시스

템 도입이 활발해지면서 각 기업내에서는 주요 기능 영역을 지원하는 여러 종류의 정보 시스템들이 독립적으로 운영되고 있어서 시스템간 단일화된 프로세스의 실행과 정보의 공유가 시급히 해결해야할 과제로 대두되고 있다. EAI (Enterprise Application Integration)는 이러한 문제에 대한 대응책으로 나타난 기술로서 Linthicum (2000)은 '기업내에서 응용 시스템이나 데이터 원천 사이에 데이터나 프로세스를 제약없이 공유하는 것' 으로 정의하고 있으며 Ruh *et al.* (2001)에서는 '미들웨어를 이용해 다양한 응용 시스템을 쉽고 빠르게 통합하는 방법을 제공하는 것' 이라고 정의하고 있다. 최근에는 기업간 정보 시스템 통합이 중요해지면서 (Olsen, 2000), 기업내 EAI와 기업간 B2B 통합을 합쳐서 응

용 시스템 통합 (Application Integration)이라는 용어로 그 범위가 확장되어 사용되고 있는 추세이다 (Linthicum, 2004). 이와 같이 최근에 기업에서의 EAI 기술의 필요성을 일깨우는 주된 동인으로는 전자 상거래의 확산, 기업 인수 합병의 증가 및 ERP 패키지의 광범위한 보급을 들 수 있다 (Erasala *et al.*, 2003).

EAI는 공유의 대상에 따라 몇 가지 유형으로 나눌 수 있는데, Ruh *et al.* (2001)에서는 사용자 인터페이스 통합 (presentation integration), 데이터 통합 (data integration), 기능 통합 (functional integration)의 3가지 유형으로 분류하였고, Linthicum (2000)에서는 데이터 수준, 응용 시스템 인터페이스 수준, 메서드 수준 및 사용자 인터페이스 수준의 4가지로 분류하였다. 그리고 Linthicum (2004)에서는 정보 지향, 서비스 지향, 프로세스 통합 지향 및 포털 지향의 4가지로 구분하였는데 이 3가지 분류를 비교하여 정리하면 <그림 1>과 같으며 최근에는 기업 프로세스 통합이 중요한 분야로 대두되고 있다.

위의 EAI 분류 중에서 데이터 수준의 통합은 가장 기본적인데 예전부터 시스템 통합의 방법으로 널리 사용되고 있는데 이 방법은 원천 시스템에서 발생한 데이터 갱신 사건을 응용 시스템의 기능(function)들을 사용하지 않고 직접 목표 시스템의 데이터 저장소에 반영하는 방법이다. 이 방법은 응용 시스템의 기능을 사용하지 않으므로 코드를 많이 변경할 필요가 없어서 단순하고 비용이 적게 드는 장점을 갖고 있다. 반면, 이 방법은 간단하고 기본적인 방법이므로 복잡한 비즈니스 규칙이 있는 응용 시스템이거나 복잡한 데이터 구조를 가지고 있는 대규모 시스템 또는 은행이나 회계와 같은 트랜잭션 기반 시스템을 위한 데이터 통합 방법으로는 적합하지 않다 (Juric *et al.*, 2002).

많은 기업에서는 이러한 데이터 수준의 통합 문제를 해결하기 위해 데이터 통합 요구가 발생할 때마다 직접 프로그램 코드를 변경하거나 혹은 데이터 통합 기능을 제공하는 상용 시스템을 도입하여 사용하게 된다. 데이터 통합 기능을 제공하는 상용 시스템으로는 BusinessWorks (Tibco, 2004)나 Business Service Platform (WebMethods, 2004)과 같이 EAI 전 범위에 걸친 기능을 제공하는 통합 솔루션과 Replication Server (Sybase, 2004)나 DB2 Information Integrator (IBM, 2004)와 같이 데이터 수준의 통합 기능을 제공하는 시스템들이 있다. 그러나 이러한 상용 시스템들은 고가이며 제공 기능이 방대하고 복잡하여 사용하기 어렵고 특정 플랫폼에 한정적이어서 시스템 사용에 요구되는 환경을 설정하기 어렵다는 단점을 갖고 있다. 특히, 중소기업의

User Interface Level	Presentation Integration	portal-oriented
		Business process Integration-oriented
Method level	Functional integration	service-oriented
Application Interface level		
Data level	Data Integration	Information-oriented

(Linthicum, 2000) (Ruh *et al.*, 2001) (Linthicum, 2004)

<그림 1> EAI 유형 분류

경우와 같이 전문 개발자가 부족한 환경에서는 이러한 상용 시스템을 이용하여 일반 사용자가 정보 시스템간 데이터 통합 요구를 만족시키기는 매우 어려운 실정이다. 이러한 문제점을 해결하기 위해서는 해당 프로그램 변경을 최소화하면서 전문 지식이 없는 일반 사용자도 쉽게 필요한 데이터의 시스템간 교환을 용이하게 하는 통합 도구의 개발이 요구되고 있다.

본 연구에서는 중소기업에서 발생하는 데이터 통합 요구를 만족시키기 위해 1) 별도의 프로그램 코딩 없이도 데이터 교환 작업이 가능하도록 하는 체계적인 데이터 교환 작업 절차를 제시하고, 2) 원천 시스템 데이터와 목표 시스템 데이터간의 관계를 명시적인 메타데이터로 관리함으로써 일반 사용자도 그래픽 사용자 인터페이스를 이용하여 데이터 교환 작업을 손쉽게 할 수 있도록 하는 데이터 통합 미들웨어의 구조 및 기능을 제시하고 이를 시스템으로 개발하여 그 유용성을 보이고자 한다.

2. 데이터 수준 EAI 방법

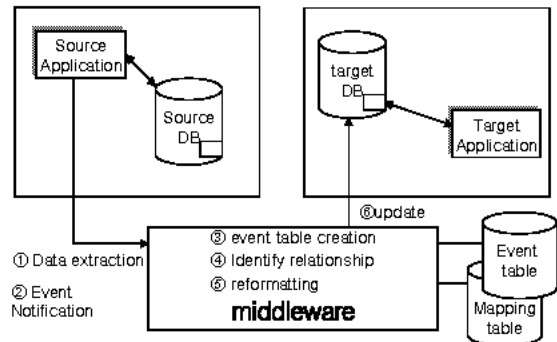
데이터 수준의 EAI를 구현하는 방법에는 크게 세 가지가 있는데, 첫째 데이터 병합 (consolidation)으로서 곳곳에 산재되어 있는 여러 데이터들을 하나의 데이터베이스로 단일화하는 것으로 데이터 웨어하우스 (DW)가 그 대표적인 예이며 데이터 통합을 위해서는 데이터 추출, 정제, 변환이나 로딩 등을 필요로 한다. 둘째는 데이터 연합 (federation)으로서 데이터들은 물리적으로 분산되어 있지만 마치 하나의 데이터베이스인 것처럼 데이터에 대한 통합된 시각을 제공한다. 마지막으로 데이터 복제 (replication)로서 원천 시스템의 데이터 저장소에 갱신 사건이 발생하면 이를 목표 시스템의 데이터 저장소에 복사하게 된다.

데이터 복제 방법은 다시 4가지로 나누어지는데, 1) SAP와 같은 대규모 상용 시스템들은 목표 시스템의 데이터를 접근할 수 있는 다양한 API

(Application Program Interface)를 제공하므로 원천 시스템에서 데이터 갱신 사건이 발생하면 이러한 API를 이용하여 목표 시스템을 갱신할 수 있다. 2) 다양한 API가 준비되어 있지 않은 시스템인 경우에는 원천 시스템에서 목표 시스템 데이터베이스를 직접 접근하여 갱신할 수 있다. 이 경우에는 원천 시스템의 프로그램을 직접 변경해야 하고 목표 시스템 데이터베이스를 직접 핸들링해야 한다는 위험부담이 있다. 3) 두 번째 방법의 위험부담을 완화하기 위해 원천 시스템에서 갱신 내용을 목표 시스템의 임시 파일이나 테이블에 저장하고, 목표 시스템에서는 별도의 프로그램을 만들어 주기적으로 임시 파일/테이블 내용을 조회하여 목표 데이터베이스에 갱신한다. 이 방법에서도 원천 시스템의 프로그램 변경에 대한 요구는 계속 존재한다. 마지막으로, 4) 원천 시스템이나 목표 시스템과는 독립적인 제3의 미들웨어에 의해 원천 시스템과 목표 시스템을 연결한다. 원천 시스템에서는 데이터 갱신 사건이 발생하면 이를 미들웨어에 알려주고 (notification) 미들웨어에서는 이 사건이 목표 시스템을 갱신해야 하는 사건인지를 판별한다. 판별 결과, 갱신 필요 사건이면 몇 가지 사전 작업을 거쳐 목표 시스템 데이터베이스를 갱신하는 구조를 가짐으로서 원천 시스템의 변경을 최소화 한다.

본 연구에서는 원천 시스템과 목표 시스템간의 데이터 교환을 위해 위에서 분류한 4번째 방법인 미들웨어를 이용한 데이터 복제 방법을 사용하는데 <그림2>에 데이터 복제 방법 절차를 간략하게 표현하고 있다.

미들웨어 방법에 의한 데이터 통합에 관한 연구로는 강진구 외 (2003)에서 PDM 시스템과 ERP 시스템을 연결하기 위해 상용 공정 계획 시스템을 중계 시스템으로 이용하였다. 이 연구에서는 자체 개발한 인터페이스를 이용하여 PDM 시스템의 EBOM (Engineering BOM)을 추출한 후 이를 이용하여 상용 공정계획 시스템에서 MBOM (Manufacturing BOM)과 공정계획서를 작성한 후 이를 ERP로 보내는 인터페이스를 개발하였다. 이 시스템에서는 데이터 교환 작업이 필요할 때마다 데이터의 추출과 목표 시스템으로의 갱신이 작업자의 수작업에 의해 이루어지고 구현된 시스템이 다른 환경에서는 적용하기 어려운 전용 시스템이라는 한계를 갖고 있다. Ou-Yang & Jiang (2002)에서는 본 연구에서와 같이 PDM과 ERP 사이의 데이터 복제를 위한 별도의 미들웨어를 개발하였으나 필요시 원천 시스템에서 발생한 갱신 사건을 사용자의 명령에 의해 검색하여 가져와야 하고 (pull type) 데이터 교환 작업시에도 사용자가 교환 명령을 시스템에 주어야 하는 등 사용자의 개입이 많은 구조를 갖고 있다.



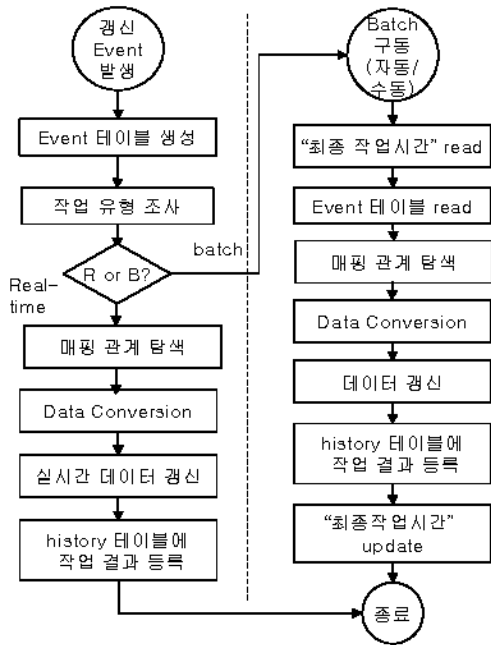
<그림 2> 미들웨어에 의한 데이터 복제 방법

본 연구에서는 원천 시스템에서 데이터 갱신 사건을 공지하면 (push type), 개발된 미들웨어에서 사전에 정의된 ‘원천 데이터-목표 데이터 매핑 관계’를 이용하여 실시간 동기적으로 목표 데이터베이스로의 갱신 작업을 수행하게 된다. 또 데이터의 종류나 사용 목적에 따라 주기적인 갱신을 필요로 하는 데이터들은 배치 작업에 의해 데이터 갱신을 할 수 있도록 작업 방법을 세분화하였다. 원천 데이터와 목표 데이터간 매핑은 데이터베이스 필드 단위로 직접 매핑하는 방법과 XML 스키마나 STEP (오유천과 한순홍, 1999)등 중간 형태의 형식으로 매핑하는 방법이 있는데 본 연구에서는 직접 매핑 방식에 의해 매핑 테이블을 사용자가 직접 등록 관리한다.

3. 미들웨어를 이용한 복제 방법에 의한 데이터 교환 절차

본 연구에서의 복제에 의한 데이터 교환 작업은 <그림3>에 나타난 바와 같이 아래와 같은 절차를 거쳐 이루어진다.

첫째, 실시간 교환인 경우에는 ① 원천 시스템에서 관심있는 데이터베이스에 갱신 행위가 이루어지면 이 사건이 미들웨어로 통보된다. ② 원천 시스템의 데이터 갱신 사건을 별도의 테이블에 저장한다. 이는 배치 갱신 작업을 위해서 또는 실시간 갱신 작업시 발생할 수도 있는 이상상황을 복원하기 위해서 필요하다. ③ 해당 사건의 교환 작업 유형이 실시간 작업인지 배치 작업인지 확인한다. ④ 발생한 갱신 데이터에 의해 복사될 목표 시스템 데이터를 원천 데이터-목표 데이터 연관 관계를 정의한 매핑 테이블을 참조하여 결정한다. ⑤ 목표 시스템으로 데이터를 복제하기 전에 데이터 변환이 필요할 경우에는 전처리 작업을 한다. 데이터 변환 작업은 데이터 형태 변환과 데이터 값 변환으로 구분된다. 본 연구에서는 데이터 형태 변환으로 숫자형 데이터에서의 실수/정수 변환과 자리 수 변환을 지원하고 시간형 데이터에서의 시간 표시 형태 변환을 지원한다. 데



<그림 3> 데이터 교환 작업 절차

이러한 값 변환은 원천 데이터와 특정 변수와의 사칙 연산을 지원한다. 데이터 값 변환을 위한 변환 규칙은 매핑 테이블에서 관리하고 값 변환에 사용되는 특정 변수들은 별도의 테이블로 유지 관리한다. ⑥ 목표 시스템 데이터베이스를 갱신한다. ⑦ 목표 데이터 필드명, 작업 성공 여부 등의 작업 결과를 이력 테이블에 기록한다.

둘째, 배치 작업인 경우에는 ① 데이터 교환을 필요로 하는 작업 패키지를 지정하여 자동 혹은 수동 방식으로 미들웨어를 구동시킨다. ② 마지막으로 배치 작업이 이루어진 시간을 확인한다. ③ 사건 테이블에 저장되어 있는 레코드 중에서 성공적으로 종료된 최종 배치 작업 시간 이후에 발생한 원천 시스템 갱신 사건들을 읽는다. ④~⑦은 실시간 작업과 동일하다. ⑧ 배치 작업의 최종 작업 시간을 기록하고 작업을 종료한다.

4. 시스템 구조 및 기능

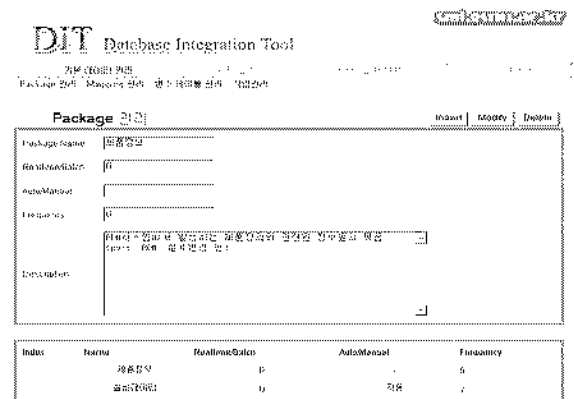
개발된 시스템의 구조는 <그림 4>에서와 같이 목표 시스템 갱신 작업에 필요한 각종 마스터 데이터를 유지 관리하는 '기본 데이터 관리 모듈' 과 데이터 추출, 재가공 및 목표 시스템 갱신 등 교환 작업을 수행하는 '데이터 교환 모듈' 과 수행되고 있는 데이터 교환 작업의 현황 및 이력을 파악하는 '데이터 교환 모니터링 모듈' 및 이상상황을 처리하는 '이상상황 관리 모듈' 의 4가지 모듈로 구성

1. 기본 데이터 관리 - 작업 패키지 등록 - 데이터 매핑 - 변수 관리 - 배치작업관리	2. 데이터 교환 - 원천 데이터 갱신 사건 통지 - 원천 데이터 갱신 사건 기록 - 목표 시스템 갱신 여부 결정 - 데이터 변환 - 목표 시스템 갱신
3. 데이터 교환 모니터링 - 데이터 교환 현황 - 데이터 교환 History	4. 이상 상황 관리 - 이상 상황 감지 - 이상 상황 처리

<그림 4> 시스템 기능 구조도

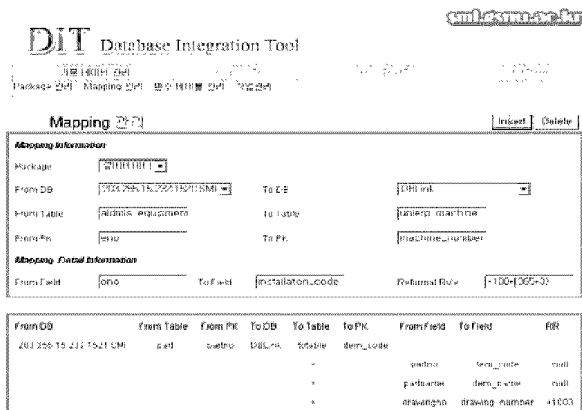
되어 있다.

기본 데이터 관리에서 작업 패키지는 데이터의 유형이나 사용 빈도 등에 따라 같은 방식으로 작업되어야 할 갱신 사건들을 그룹핑하여 하나의 작업 단위를 구성하는 것을 의미한다. 즉, 작업 패키지는 실시간 교환인지 배치 교환인지를 구분하는 작업 구분과 배치 작업인 경우에 자동 수행인지 수동 수행인지를 구분하는 작업 형태 및 배치 작업 실행 주기 등을 그 속성으로 갖는다. <그림 5>에 작업 패키지를 등록하는 화면을 나타내고 있는데 화면 상단이 작업 패키지를 등록하는 화면이고 화면 하단이 등록된 패키지 들을 전체적으로 조회하는 화면이다.



<그림 5> 작업 패키지 등록 화면

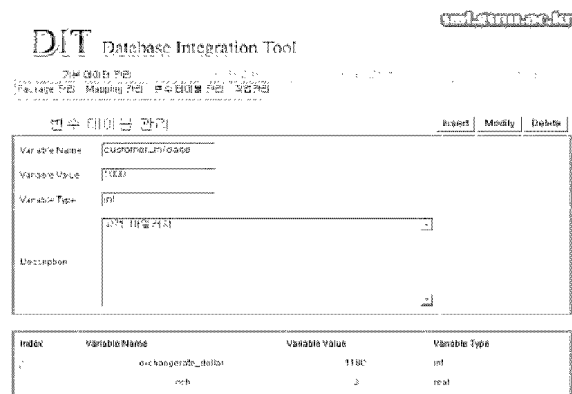
데이터 매핑은 가장 중요한 기본 데이터로서 원천 데이터-목표 데이터 연관 관계를 정의한다. 즉, 원천 시스템의 DB명, 테이블명, 필드명을 목표 시스템의 해당 이름과 매핑시키고 데이터를 재가공하는 변환 규칙을 정의한다. 데이터 매핑시 하나의 교환 작업 단위로 관리되어야 할 필요성이 있는 데이터들은 동일한 작업 패키지로 등록한다. <그림 6>이 데이터 매핑 정의를 등록하는 화면인데 화면 상단에서 테이블 이름, 주 키 이름, 필드 이름 등 원천 시스템 (From DB)과 목표 시스템 (To DB)간의 연관 관계를 등록하고 데이터 변환이 필요한 경우에는 변환 규칙



<그림 6> 데이터 매핑 정의 등록 화면

(reformat rule)을 정의한다. 화면 하단은 테이블 단위로 정의된 필드간의 매핑 관계를 보여 준다.

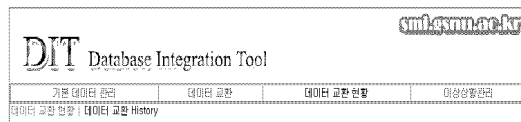
변수 관리는 환율, 무게/길이 변환 단위 등 원천 데이터의 값 변환에 사용되는 변수 이름과 그 값을 등록 유지한다. 배치 작업 관리는 데이터 교환 작업을 사용자가 수작업으로 지정하여 작업을 수행하는 경우에 작업 패키지를 선택하여 이를 구동시키는 기능을 제공한다. <그림 7>에 변수를 등록하는 화면을 보여준다.



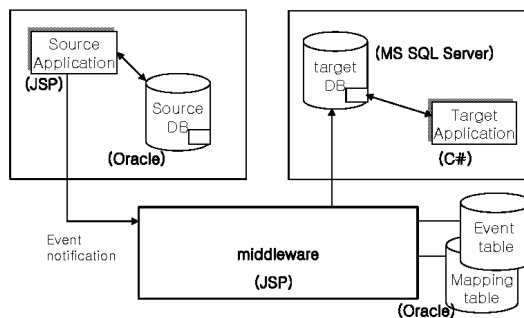
<그림 7> 변수 관리 화면

데이터 교환 모듈은 시스템 내부에서 사용자와 독립적으로 수행되는 모듈로 원천 데이터 갱신 사건이 발생하여 데이터 통합 미들웨어 측에 통지되면 갱신 사건을 별도의 이벤트 테이블에 기록하고 이 갱신 사건이 변경시키는 목표 시스템의 데이터를 매핑 테이블을 참조하여 결정하고 데이터 변환이 필요한 경우에는 매핑 테이블에 정의된 변환 규칙을 참조하여 데이터 형태 변환이나 데이터 값 변환을 실시한다. 그 후, 목표 데이터베이스에 접속하여 해당 데이터를 갱신한다.

데이터 교환 모니터링에서는 현재 수행되고 있는



<그림 8> 데이터 교환 이력 화면



<그림 9> 데이터 통합 미들웨어 테스트 환경

교환 작업의 현황을 모니터링 할 수 있고 교환 작업의 이력을 여러 가지 형태로 조회 할 수 있다. 이상 상황 관리 모듈에서는 이상상황이 감지되면 사용자에게 메시지로 알려거나 시스템내에서 복구 작업을 수행하는 기능을 수행한다. <그림 8>에서는 특정 기간 사이에 발생한 목표 시스템 갱신 이력을 레코드 단위로 패키지 이름, 작업 구분, 목표 DB 이름, 테이블 이름, 갱신이 일어난 레코드 주 키 값, 갱신 작업이 정상인지 이상상황으로 종료했는지를 나타내는 종료 구분과 발생한 갱신 이벤트가 삽입 (CI), 변경 (CH), 삭제 (CD) 중 어떤 유형인지를 나타내는 이벤트 유형 및 갱신 시간을 표시하고 있다.

개발된 데이터 통합 미들웨어는 JSP와 오라클 데이터베이스를 이용하여 개발하였는데 개발된 시스템의 작동성을 보이기 위해 <그림 9>와 같은 환경에서 테스트를 실시하였다. <그림 9>에서와 같이 원천 시스템은 JSP와 오라클 데이터베이스로 이루어진 웹 기반 PDM 시스템으로서 JSP로 작성된 응용 프로그램에서 데이터 갱신이 이루어지면 PDM 데이터베이스 (오라클)에 저장되며, 이 갱신 사건을 데이터 통합 미들웨어에 알린다. 미들웨어에서는 데이터 교환에 필요한 매핑 정보나 데이터 갱신 사건들을 오라클 데이터베이스에 저장 관리한다. 미들웨어에서는 통

보관은 데이터 갱신 사건에 대해 작업 유형에 따라 교환 작업을 실시하고 작업 결과는 목표 시스템의 데이터베이스에 반영된다. 목표 시스템은 C#과 MS SQL 서버 데이터베이스를 사용하여 개발되었다.

5. 결론 및 추후 연구 과제

본 연구에서는 하나의 조직내에서 정보의 공유나 실시간 의사 결정 지원을 위해 데이터 원천에서 특정 데이터를 추출하여 재가공하여 목표 시스템에 갱신해야 할 필요성이 있을 때 이를 미들웨어를 사용하여 데이터를 효과적으로 교환하는 체계적인 절차를 제시하고 이를 시스템으로 구현하였다. 기존에는 원천 시스템 데이터와 목표 시스템 데이터간의 관계를 소스 프로그램내에서 관리하였는데 본 연구에서는 이를 명시적으로 추출하여 메타데이터로 관리함으로써 프로그램 변경을 최소화하였다. 그리고 데이터 교환을 위한 그래픽 사용자 인터페이스를 제공함으로써 프로그램에 대한 전문 지식이 없는 사용자도 비교적 손쉽게 시스템을 운영할 수 있게 하였다.

따라서 본 연구에서 개발한 시스템은 전문 개발자가 부족하고 고가의 상용 EAI 시스템을 구입하기 어려운 중소기업의 정보 시스템간 데이터 교환 요구를 효과적으로 지원할 수 있다.

그러나 본 연구에서 개발된 시스템은 아직 프로토타입 수준이어서 실제적으로 기업에서 사용하기 위해서는 보완해야 할 점이 많이 있다. 첫째, 개발된 미들웨어가 플랫폼 독립적으로 운영되려면 미들웨어와 원천/목표 시스템 사이에 느슨한 결합이 필요한데 이를 위해서는 웹 서비스 기능의 채택이 요구된다. 둘째, 테스트 환경에서는 자바와 C# 언어로 작성된 정보 시스템을 대상으로 하였는데 실제적으로 기업 환경에서 운영되려면 COBOL, C, Power Builder와 같은 언어로 작성된 레가시 시스템을 데이터 통합 미들웨어와 연결하기 위한 웹 서비스 구현에 관한 연구가 추가적으로 수행되어야 한다. 마지막으로 데이터 변환 작업에 있어서 현재는 단일 데이터 필드에 변수 값을 사칙 연산하는 수준에 머무르고 있는데 복수 개의 데이터 필드로 구성된 복합 변환 규칙을 다루기 위해서는 별도의 규칙 엔진을 이용하여 변환 규칙을 독립적으로 관리하는 방안 에 대한 연구도 필요하다.

후기

본 연구는 산업자원부 지역특화·중기거점기술 개발사업 (항공기 기체 지능형 통합 생산 시스템 및 고속 가공기 개발)의 지원으로 수행되었음.

참고문헌

1. 강진구, 한판희, 김정진, "공정계획 시스템을 이용한 PDM과 ERP 시스템 연동 환경 개발," 한국경영과학회/대한산업공학회 2003 춘계 공동학술대회, pp.63-69, 2003.
2. 오유천, 한순홍, "CAD와 PDM 시스템간의 제품 구조 정보 매핑," 1999 한국 CAD/CAM 학회 학술 발표회 논문집, pp.244-248, 1999.
3. Erasala N., Yen D. C. and Rajkumar T. M., "Enterprise Application Integration in the Electronic Commerce World," Computer Standards & Interfaces, Vol.25, No.2, pp.69-82, 2003.
4. IBM, IBM DB2 Information Integrator, www.ibm.com.
5. Linchicum D. S., Enterprise Application Integration, Addison Wesley, Reading, Massachusetts, 2000.
6. Linchicum D. S., Next Generation Application Integration, Addison Wesley, Boston, 2004.
7. Juric M. B., Basha S. J., Leander R. and Nagappan R., Professional J2EE EAI, Wrox Press, Birmingham UK, 2002.
8. Olson, G., An Overview of B2B Integration, EAI Journal, 28-36, May, 2000.
9. Ou-Yang C. and Jiang T. A., "Developing an Integration Framework to Support the Information Flow Between PDM and ERP," International Journal of Advanced manufacturing Technology, Vol.19, No.2, pp.131-141, 2002.
10. Ruh W. A., Magnnis F. X. and Brown W. J., Enterprise Application Integration, John Wiley & Sons, New York, 2001.
11. Sybase, Sybase Replication Server, www.sybase.com/products/middleware, 2004.
12. Tibco, TIBCO BusinessWorks, www.tibco.com, 2004.
13. Webmethods), Enterprise Services Platform, www.webmethods.com, 2004.