
XMDR 데이터 허브 기반의 Proxy 데이터베이스를 이용한 데이터 상호운용 프레임워크

문석재* · 정계동* · 최영근*

Data Interoperability Framework based on XMDR Data Hub using Proxy DataBase

Seok-Jae Moon* · Gye-Dong Jung* · Young-Keun Choi*

본 연구는 2008년도 광운대학교 교내연구비 지원에 의해 연구되었음.

요 약

본 논문에서는 XMDR(eXtended Meta-Data Registry) 데이터 허브 기반의 Proxy Database를 이용하여 Legacy Database간의 데이터 상호운용이 가능한 프레임워크를 제안한다. 협업 환경에서는 Legacy Database간의 상호운용을 하는데 있어서 데이터의 구조, 의미, 형식상의 이질적인 문제들이 발생한다. 또한 실시간으로 변화하는 데이터를 종류와 형식에 관계없이 지속적으로 일관성을 유지하기가 어렵다. 본 논문에서는 XMDR 데이터 허브를 이용하여 Legacy DB간의 데이터 통합 및 상호운용에서 발생할 수 있는 이질적인 문제를 해결한다. Proxy Database를 이용하여 상호운용하고자 하는 데이터들이 종류와 형식에 상관없이 호환이 가능하고, 지속적으로 정확한 정보를 실시간으로 일관성 있게 제공하는 프레임워크를 제안한다.

ABSTRACT

We propose Framework that (should) have data interoperability between Legacy DataBases using Proxy DataBases based on XMDR(eXtended Meta-Data Registry) Data Hub in this papers. It may occur some problems among data structure, semantics and other heterogeneous problems between interoperability of legacy DB on cooperation environment. Also, It is hard to keep consistency of Data that changes on realtime, regardless of data variety and type. In this paper, Using XMDR data hub, we solve the problem that was occurred by data integration and interoperability between legacy DB. We suggest the framework which are compatible with any class and type of interoperability-data and offer accurate information with consistency in real-time using proxy database.

키워드

XMDR(eXtended Meta-Data Registry), Collaboration, Distributed DataBase

I. 서론

IT환경이 복잡해짐에 따라 기업 내의 분산된 어플리

케이션의 정보 활용이 필요하게 되었으며, 기업 간의 M&A 및 아웃소싱, 협력 등의 과정에서 각 기업들의 사용하던 데이터베이스에 대한 데이터 상호운용이 크게

필요하게 되었다[1]. 일반적으로 기업 내의 어플리케이션의 필요한 정보들은 분산되어 있어 데이터 상호운용을 하는데 데이터들이 구조적, 의미적으로 다르게 표현되어 있어 이질적인 문제가 발생하더라도 데이터 상호운용이 불가피하다[2]. 이런 데이터 상호운용의 목적은 기업의 조직과 주요 업무, 핵심 어플리케이션으로부터 발생하는 데이터 소스들의 표준 규칙과 메타데이터를 이용하여 중복성을 제거하고, 오직 단일 데이터를 정확하게 제공하는데 있다. 특히 이기종시스템마다 다양한 어플리케이션에서 나오는 대량의 데이터들이 종류와 형식에 상관없이 호환이 가능하고, 지속적으로 정확한 정보를 실시간으로 일관성 있게 제공해야 하는 것이 관건이다[1]. 따라서, 본 논문에서는 분산되어 있는 Legacy DB들이 독립성을 유지하면서 하나의 인터페이스를 제공할 수 있도록 상호운용에서 발생하는 데이터들의 불일치(스키마 변환, 매칭, 매핑-스키마 이질성)을 해결하는 프레임워크를 제안한다.

본 논문에서 제안한 프레임워크는 데이터허브, 미디어이터, Legacy DB으로 구성된다. 데이터허브의 XMDR은 스키마 변환, 매칭, 매핑을 해결하는 역할을 그리고 Proxy DB는 통합된 데이터를 집중시키는 역할을 한다. 미디어이터는 Legacy DB에서 데이터를 검색할 수 있는 SQL Query 언어를 생성하는 역할을 한다. 본 논문에서 제안한 프레임워크는 크게 4가지 범주의 서비스 영역으로 분리된다. 첫 번째, 데이터의 적시성을 제공하여 기업 내,외부에 분산되어 있는 서로 다른 시스템들을 하나의 인터페이스를 통해서 데이터를 공유할 수 있도록 하는 환경을 제공한다. 두 번째, 데이터 허브의 핵심 역할로서 단일 뷰어를 제공하여 기업 정보의 전사 컨센서스를 확보한다. 다시 말해 동일한 의미와 동일한 표현 형식으로 제공된다는 것이다. 세 번째, 허브-앤-스포크라는 정보 흐름의 양방향성을 제공한다. 네 번째, 체계화된 규칙 기반의 데이터 정제, 관리를 제공한다. 즉, 정보의 전사 공유 환경을 제공하기 위해서는 체계화되고 자동화로 관리되어야 한다.

본 논문의 구성은 다음과 같다. 2장은 XMDR의 정의, 구성에 대해서 기술한다. 3장은 제안 프레임워크의 구성에 대해서 기술한다. 4장은 시스템 구현에 대해 기술한다. 5장은 타 시스템과의 비교 평가 및 성능 평가를 한다. 마지막 장은 본 논문의 결론과 향후연구에 대해 기술한다.

II. XMDR

2.1 XMDR(eXtended MetaData Registry)

본 논문에서의 XMDR은 데이터 상호운용에 따른 데이터 이질성을 해결하기 위한 XML기반의 관계형데이터베이스 메타데이터를 객체지향 데이터베이스에 저장하는 기술과 분산된 데이터의 이질성을 해결하고자 MDR과 온톨로지 시소러스를 결합한 것이다[5][6][7][8][9][10].

XMDR은 ISO/IEC 11179[3]에서 제안한 정보 공유 및 교환을 위한 표준으로 현재 많은 프로젝트가 진행 중에 있다. 특히, ISO/IEC 11179-3에서는 공유 데이터의 관리를 위한 메타 모델, 기본 속성이 제시되어있는데, 메타 모델은 의미적인 내용과 분산된 환경하의 사용자들이나 정보처리 시스템간의 공유되는 데이터 요소의 구문을 위한 표준과 안내를 제공하고 있다. 본 논문에서 제안한 프레임워크에 적용된 XMDR은 협업환경에서의 분산 Legacy DB들의 메타데이터 스키마정보를 기반으로 메타-시멘틱 온톨로지 (MSO:Meta-Semantic Ontology), 메타-로케이션(ML:Meta-Location), 인스턴스-시멘틱 온톨로지(InSo:Instance-Semantic Ontology)로 구성하였다. 각 요소는 다음과 같이 정의되고 구성된다.

- 메타-시멘틱 온톨로지: 분산된 Legacy DB들의 스키마 정보들을 시소러스화하여 표준에 맞추어 매핑한 것으로, 메타데이터의 관계성과 이질성 해결을 목적으로 한다.
- 메타-로케이션: MSO와 연계하여 Legacy DB들의 물리적인 정보인 위치정보, 접근 권한 등을 관리한다.
- 인스턴스-시멘틱 온톨로지: 실제 상호운용되는 데이터 값과 분산 데이터 값들간의 연관관계성을 시소러스화하여 정의한 것이다.

위와 같이 정의된 요소들은 ISO/IEC 11179-3에서 제안한 데이터의 속성 명세를 따른다. 데이터의 기본 속성은 식별속성, 정의속성, 관계속성, 표현속성으로 명세는 다음과 같다.

- 식별 속성(Identify Attribute): 데이터 요소의 식별을 위한 속성.

표 1. 메타데이터 스키마간의 이질성분류
Table 1. Classifying Heterogeneity of Metadata Schema

분류	이질성 유형	설명	예(Example)
구조	합성(Composition)	다수의 스키마 요소가 하나의 스키마 요소에 합으로 표현	(FirstName U LastName) -> Name
	분해(Decomposition)	하나의 스키마 요소가 다수의 스키마 요소로 분리되어 표현	Name -> (FirstName U LastName)
	재배치(Rearrangement)	스키마 요소의 순서 정보가 필요하며 그 배치가 다른 표현	FirstName-LastName -> LastName-FirstName
의미	대체(Substitution)	스키마 요소간의 동등한 의미의 스키마 요소로 표현	Name -> author, writer, fullname
표현	코드형식(Code Format)	코드형식의 다른 표현	YYYY-MM-DD <-> MM-DD-YYYY
	단위형식(Unit Format)	단위형식의 다른 표현	Pound <-> kilogram Mile <-> Kilometer,
	데이터형식(Data Format)	데이터형식의 불일치 표현	VARCHAR(20) <-> VARCHAR(50) Integer <-> Float

- 정의 속성(Define Attribute): 데이터의 요소의 의미를 갖는 속성.
- 표현 속성(Presentation Attribute): 데이터 요소의 표현방식에 따른 속성.

본 논문에서 제시한 XMDR은 위와 같은 명확한 요구사항을 만족하여 구성된 것으로 효율적으로 데이터의 상호운용을 수행할 수 있다.

XMDR은 분산 데이터의 자율성과 독립성을 보장할 수 있다. 또한 기본 DB들의 스키마의 재구성이나 변경 없이 사용이 가능하다. XMDR의 자세한 구성내용은 2.2, 2.3, 2.4에서 기술한다.

2.2 메타-시맨틱 온톨로지

MSO(Meta-Semantics Ontology)는 메타데이터 상호운용에서 발생할 수 있는 동등한 의미를 갖는 메타데이터 간에 일어날 수 있는 문제를 분류하고 그 유형을 정리하였다. 표1은 동등의미 메타데이터 간에 발생할 수 있는 구조, 의미, 표현의 불일치 유형을 요약한 것이다. 표1에서 분류한 불일치 유형들은 단독으로 존재할 수도 있지만 일반적으로 여러 유형들이 혼합되어 나타난다[4].

2.3 메타-로케이션

MLOC(Meta-Location)는 분산 Legacy DB들의 위치정보, 접근 권한 정보를 MSO와 릴레이션하여 Legacy DB들로부터 검색이나 트랜잭션을 하기 위한 컨넥션할 때

표 2. 인스턴스 값들의 연관성
Table 2. Relations of Instance Values

연관 유형	설명	예(Example)
동등관계 (Equal-Relation)	데이터를 표현하는 방식은 다르지만 의미상 동등한 의미를 가지는 관계.	Computer = Com = 컴퓨터
포함관계 (Contain-Relation)	데이터를 의미하는 것에 포함관계.	본체 -> Memory, CPU
부분관계 (Part-Relation)	데이터간의 의미상 상위, 하위 개념관계	전자제품 - 컴퓨터 - 서버
참조관계 (Reference-Relation)	데이터간의 의미상 참조가 되는 관계	PC -> Monitor, Keyboard

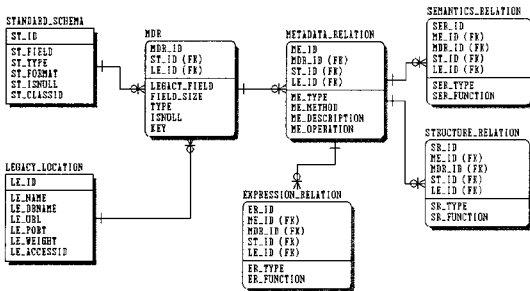
필요한 정보를 제공하는 역할을 한다.

2.4 인스턴스-시맨틱 온톨로지

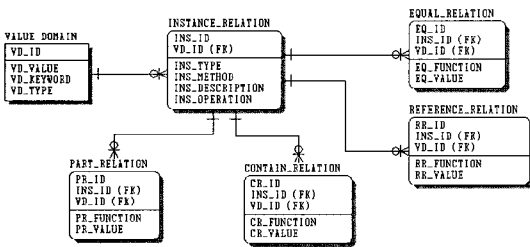
InSO(Instance-Semantics Ontology)는 협업 환경에서 데이터 값이 상호운용할 때 실제 값(value)들 사이에 연관성을 구조화하여 구체적으로 관계성을 정의한 것이다. 단순한 검색이 아니라 검색 키워드에 대한 의미성, 유사성을 고려하여 검색에 대한 결과를 확장하려는 데 목적이 있다. 표2는 값들 간의 연관관계를 4가지로 분류하였다. 예를 들어, ‘Computer’에 대해 요청한다면 ‘COM’, ‘컴퓨터’ 등과 같은 의미는 갖지만 다른 표현으로 된 데이터도 같이 검색되어 Computer에 대한 정보를 더욱 확대하여 검색할 수 있도록 한다.

2.5 XMDR 스키마 설계

XMDR 설계는 2.3, 2.4, 2.5에서 기술한 방법으로 이질적인 문제를 해결하고 데이터 상호운용의 일관성을 위해 표준 스키마 항목을 결정한다.



(a) 메타-시맨틱 온톨로지 및 메타-로케이션 ERD
(a) Meta-Semantic Ontology and Meta-Location ERD



(b) 인스턴스-시맨틱 온톨로지 ERD
(b) Instance-Semantic Ontology ERD

그림 1. XMDR ERD 구성
Figure 1. XMDR ERD Composition

III. 제안 프레임워크

3.1 시스템 개요

본 논문에서 제안한 프레임워크의 목적은 Legacy DB들이 독립성을 유지하면서 하나의 인터페이스를 제공할 수 있도록 데이터 상호운용에서 발생하는 이질성을 해결할 수 있도록 설계한 것으로 다음 그림2와 같다. 프레임워크 구성은 Client Applications, DataHub, Mediator, Legacy Systems(DB)로 구성된다.

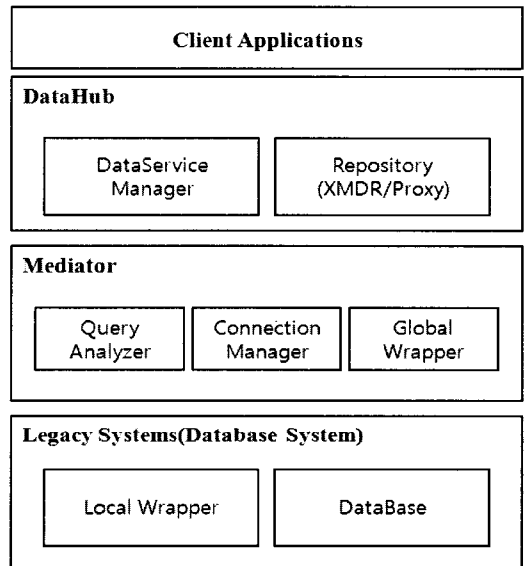


그림 2. 데이터 상호운용 프레임워크
Figure 2. Data Interoperability Framework

3.2 구성요소

3.2.1 Client Applications(CA)

CA는 협업 환경에서 특정 서비스를 제공하기 위해 개발된 프로그램 또는 Legacy System이다.

분산되어 있는 데이터를 통합적으로 서비스하기 위한 것이 목적이므로 단일 뷰어 인터페이스로 구성되거나, 다른 Legacy DB에서 운용중인 데이터를 필요로 하는 시스템들이다.

3.2.2 데이터허브

데이터허브는 CA와 Mediator사이에서 중계자적인 역할을 수행하는 요소이다. 주된 역할은 표준 항목과 분산

Legacy DB의 메타데이터 스키마 정보를 관리하고, 스키마간의 변환, 매칭, 매핑에 필요로 하는 정보를 제공한다. 그리고 Proxy DB에 수집되어 있는 데이터를 CA에 전달한다.

3.2.3 미디어이터

미디어이터는 데이터허브에서 넘어온 질의 정보와 스키마 정보를 QA(Query Analyzer)에서 분석하여 Legacy DB에 맞는 SQL Query 언어를 생성하고 전송하는 역할과 각 Legacy DB에서 수집되어온 결과를 저장할 수 있는 Proxy DB에 테이블을 생성하는 역할을 한다. 또한 각 Legacy DB에서 수집되어 넘어온 데이터를 파싱하여 Proxy DB에 insert한다.

3.2.4 Legacy Systems

Legacy DB는 협업 환경에 상호운용되는 데이터를 독립적으로 운용하고 있는 기존 시스템들이다. CA에 필요한 데이터를 제공하거나, 필요한 데이터를 다른 협업된 Legacy DB에 요청하기도 한다.

3.3 프레임워크 프로세스

그림3은 본 논문에서 제안한 프레임워크에서 데이터를 상호운용하는 프로세스 전체 흐름이다. CA가 DSM

(DataService Manager)에 데이터를 요청하면 DSM은 XMDR에 메타데이터 스키마 정보를 검색하여 CA가 요청한 데이터와 스키마 정보를 미디어이터에게 전달한다. 그 다음 미디어이터의 QA는 넘겨받은 스키마 정보를 이용하여 Proxy DB에 생성할 테이블 SQL Query를 CM(Connection Manager)에게 넘겨줌과 동시에 CA에 요청한 데이터를 Legacy DB의 스키마에 맞는 SQL Query 언어를 Global Wrapper에게 넘겨준다.

그 다음 Global Wrapper는 SQL Query 언어를 XML 형태로 변환하여 Local Wrapper들에게 전송한다. 각 Legacy의 Local Wrapper는 XML형태의 SQL Query를 파싱하여 DB에 트랜잭션을 한다. 각각 트랜잭션된 결과 데이터는 다시 Local Wrapper에 의해 다시 XML 형태로 Global Wrapper에 전송되고 반환된 결과 데이터를 분석하여 CM에 전달하면 CM은 생성된 Proxy에 Table에 Insert된다. 그 다음은 DSM가 Proxy에 결과가 insert된 table을 검색하여 최종 결과 데이터를 CA에게 Response 한다.

3.4 SQL Query 프로세싱

SQL Query는 CA의 데이터 요구에 따른 표준 항목, 조건, 접근위치와 권한을 포함하여 생성된다. 그림4와 같이 MSO에서 분산된 DB에 맞는 스키마 정보, ML에서 위

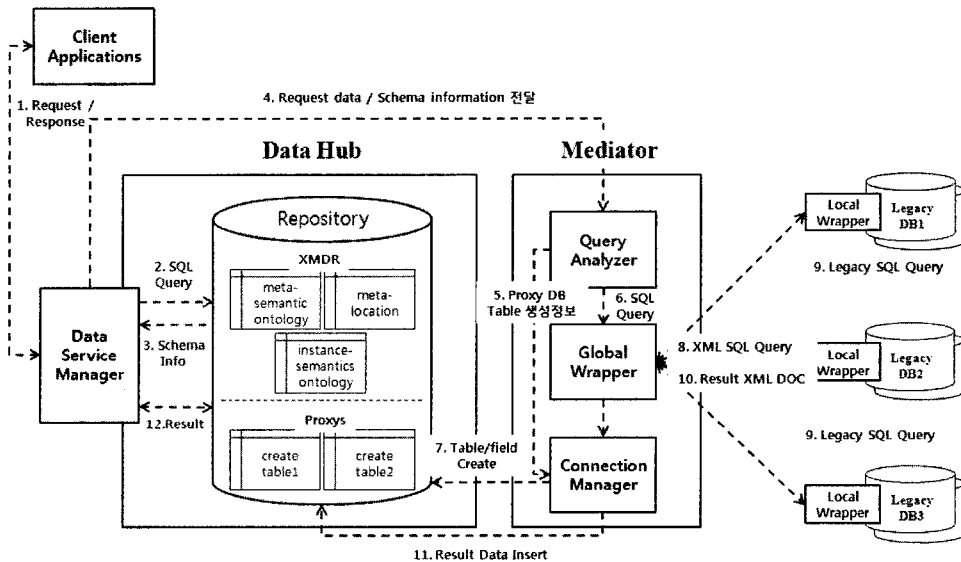


그림 3. 프레임워크 흐름도
Figure 3. Framework Flow

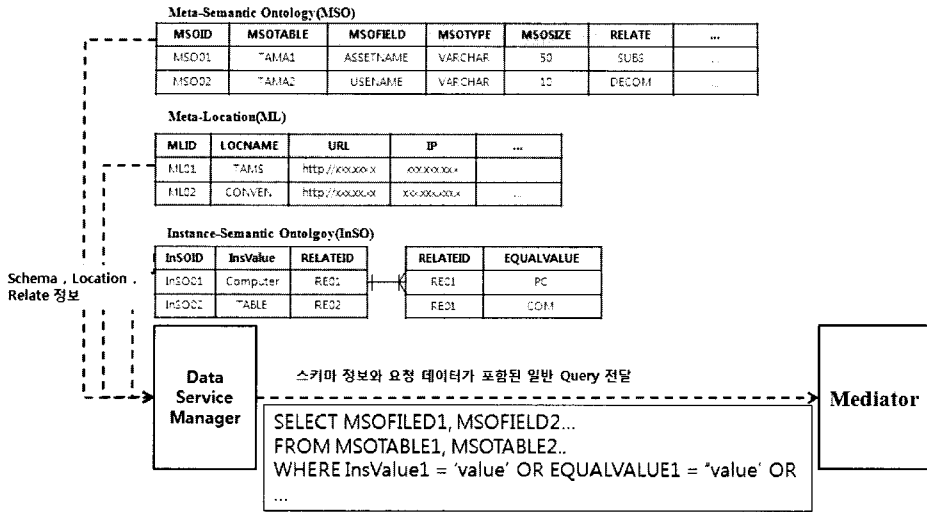


그림 4. SQL Query 프로세싱 과정
Figure 4. SQL Query Processing

```

<?xml version="1.0" encoding="euc-kr"?>
<LOCATION>
  <Legacy locid="001" url="http://www.assettask.co.kr" ip="128.134.42.5" id="tam" pass="$dfsklf">
</LOCATION>
<SQL>SELECT</SQL>
<FIELDS>
  <column cid="100002">ASSETNAME</column>
  <column cid="100003">USERNAME</column>
  <column cid="100004">USEDEPT</column>
  <column cid="100005">PRICE</column>
  ....
</FIELDS>
<FROM>
  <JOIN>INNER</JOIN>
  <table tid="20001" idref="20002" pk="tid" fk="">TAM1000</table>
  <table tid="20002" idref="20001" pk="" fk="tid">TAM2000</table>
</FROM>
<WHERE>
  <OR>
    <column cid="100002" rel="">computer</column>
    <column cid="100002" rel="">com</column>
    <column cid="100002" rel="">pc</column>
  </OR>
  ....
  <column cid="100005" rel="like">750000</column>
</WHERE>
  ....
    
```

그림 5. XML 형태의 SQL Query
Figure 5. SQL Query of XML Document

치정보, InSO에서는 요구조건에 따른 인스턴스 값에 연관정보를 DSM에서 Mediator에 전달한다. 그림5는 Global Wrapper에서 각 Legacy DB에 전송할 XML로 변환된 SQL Query의 한 예이다.

각 태그 요소들은 다음과 같다. <LOCATION>는 Legacy DB에 물리적인 접근 정보와 접근 권한 요소이다. 하위 요소인 <Legacy>에 속성으로 locid, url, ip, id, pass가 있다. <SQL>는 SQL Query에서의 데이터 트랜잭션을 나타내는 요소로서 SELECT, INSERT, UPDATE, DELETE가 이에 해당한다. <FIELDS>요소는 SQL Query에서 "SELECT ASSETNAE, USERNAME FROM.."의 필드명을 의미한다. 하위 요소로는 <column>이 있고, 속성 cid가 있다. <FROM>는 SQL Query에서 테이블을 나타내는 요소이다. 하위 요소에 <JOIN>는 관계형 DB에서 테이블간의 릴레이션되는 관계를 의미하는 요소로 inner, outer, cross가 이에 해당한다. <table>요소 속성에 idref는 column간의 릴레이션 관계는 나타내고 pk, fk는 테이블간의 릴레이션이 되는 실제 키 값이다.

<WHERE>은 조건 값을 의미하는 요소이다. 하위노드인 <OR>는 실제 여러 값에 대한 의미이다. 그리고 속성인 rel은 값이 매칭을 의미한다.

IV. 시스템 구현

본 프레임워크 구현은 Visual Studio.NET(C#), MS-SQL2000, Oracle 8i, Windows 2003 Server를 이용하여 수행되었으며, 통합 자산 관리 시스템을 적용 사례로 하였다. 그림6은 사용자가 비즈니스 프로세스를 저장-프로시저 레포지토리에 연결하여 자산신청, 구매, 등록, 발주, 회계에 대한 처리 과정을 프로세스별로 선택하여 시퀀스를 정의하고 그에 따른 질의가 생성되는 화면이다. 그림6에서 생성된 SQL Query는 XMDR 데이터 허브에 전달된다.

그림7에서처럼 그림6에서 전달된 SQL Query는 사용자가 요구조건을 선택, 입력하고 분산 Legacy DB를 선택한다. 그 다음 데이터허브의 레포지토리의 위치와 미디어데이터의 사용되는 컴포넌트를 입력한다. 그런 후 분산된 Legacy DB에서 상호운용 할 데이터의 저장-프로시저를 선택하고, 이를 XML 형태로 변환하게 된다. 변환된 XML문서인 SQL Query를 Execute하게 되면 Result

Proxy에 테이블이 생성되고, 요구사항에 맞는 데이터 결과가 insert된 화면이다.

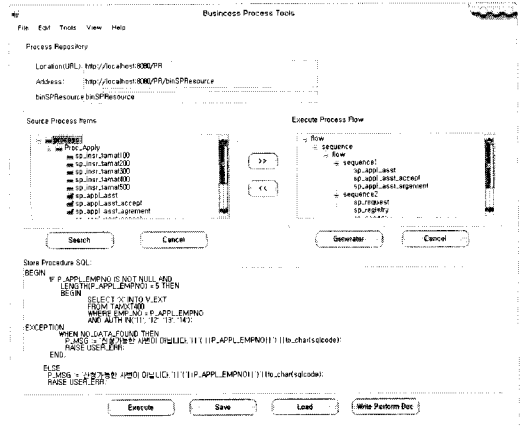


그림 6. SQL Query 처리 과정 및 생성 화면
Figure 6. SQL Query execution and create

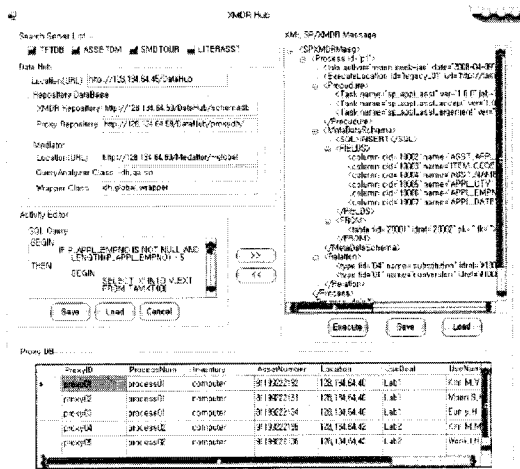


그림 7. SQL Query를 XMDR 데이터 허브에서 수행한 결과 화면
Figure 7. SQL Query from XMDR data hub execute result interface

V. 비교분석 및 성능평가

5.1 타 시스템 비교분석

본 논문에서 제안한 프레임워크는 구조 측면에서 XMDR 지원여부, 데이터 교환 자동화 여부, 저장소 사용 여부, 저장소 구조, 질의 형식 등 5가지 사항을 비교하였다. 표3에서 비교한 것과 같이 Grid-DBMS, Oracle Grid의 구조는 협업된 Legacy DB들이 실제 사용되는 스키마 정보를 기반으로 구성되어 있다. 따라서 SQL Query 기반으로 데이터를 교환할 때 데이터 구조상의 이질성은 극복 가능하지만 의미, 형식상의 이질성은 극복하기가 어렵다. 하지만 본 논문에서 제시한 프레임워크는 XMDR을 이용하여 Legacy DB들의 메타데이터 스키마 정보를 참조하여 데이터 교환할 때 발생하는 구조, 의미, 형식상의 이질성을 극복할 수 있다. 그리고 본 프레임워크는 데이터 저장소 기반의 Proxy DB를 사용함으로써 데이터 통합에 대한 유연성을 보다 용이하다. 하지만 Grid-DBMS와 Oracle Grid는 Proxy DB가 없이 데이터 상호운용을 하기 때문에 데이터에 대한 정보가 휘발성과 같이 매번 데이터를 교환해야 하는 문제가 있다. 또한 저장소 구조 측면에서는 Grid-DBMS는 분산 형태의 구조이고, Oracle Grid는 중앙 집중 구조이다. Grid-DBMS는 분산 구조이기 때문에 데이터 트랜잭션이 발생할 때 부하가 줄어들기 때문에 데이터 상호운용에 매우 유용하지만, 각 Legacy DB들이 데이터에 대한 변경이 발생했을 시에는 데이터에 동기화가 어렵다. Oracle Grid는 중앙 집중식 구조이기 때문에 상호운용에 필요한 데이터 동기화시에는 매우 유리한 구조를 갖지만, 트랜잭션이 빈번할 경우에는 과부하에 의한 문제가 발생할 수 있다. 이에 본 논문에서 제시한 프레임워크는 중앙구조와 분산 구조를 동시에 이용함으로써 빈번이 발생하는 데이터 교환은 중앙 구조식으로 해결하고, 그렇지 않은 경우

에는 분산 구조식으로 해결이 가능함으로 이를 해결할 수 있다.

마지막 항목인 질의 형식은 본 프레임워크에서는 글로벌 질의 구조의 메시지 형태로 데이터 상호운용을 한다. 이는 협업된 Legacy DB들이 독립적으로 운영하다가 메타데이터 스키마 구조에 대한 변경이 있을 시 XMDR에 있는 메타데이터 스키마 정보만 변경함으로써 매우 유연하게 대처할 수 있는 반면에, Grid-DBMS는 매번 스키마의 구조 변경이 있을 때 다른 협업된 Legacy DB들이 스키마가 변경된 정보를 적용해야 하는 어렵이 따른다.

5.2 성능평가

본 논문에서 제안한 프레임워크를 적용한 통합 자산 관리 시스템에서 데이터 접근 서비스와 수집 서비스를 수행한 결과를 바탕으로 성능을 평가한다. 시스템에서 이벤트와 상태 변화를 감지하는 서비스는 레거시 시스템의 상태와 성능에 따라 다르므로 분석 대상에서 제외한다. 통합 자산 관리 시스템에서 제공하는 데이터 접근 서비스와 수집 서비스는 기본적인 서비스로서 데이터 상호운용을 지원한다. 데이터 접근 서비스는 Legacy DB에서 발생하는 데이터 변경 이벤트에 대하여 통합 자산 관리 시스템에서 이벤트를 반영하기 위하여 제공되는 서비스이다. 또한 데이터 수집 서비스는 통합 자산 관리 시스템에서 각 Legacy DB에 위치한 데이터를 수집하여 통합 관리를 지원하기 위한 서비스로서 이질적인 데이터에 대하여 수집할 수 있다. 이러한 서비스들이 수행하는 성능에 따라 레거시 시스템들의 협업이 원활하게 이루어질 수 있다. 다음 그림8은 데이터 검색 서비스를 수행하기 위하여 미디어이터에서 SQL Query 처리시간에 대한 성능을 측정된 것이다. 연관성에 따라 의미적으로 연관된 데이터에 대한 질의 조건은 증가하게 되어 작업 수행 시간은 크게 증가하였다.

표 3. 관련 시스템과의 비교분석
Table 3. The comparison of frameworks

	Grid-DBMS	Oracle Grid	본 프레임워크
XMDR 지원여부	지원하지 않음	지원하지 않음	지원함
데이터 교환 자동화 여부	지원함	일부 지원함	지원함
저장소 사용여부	사용안함	사용안함	사용함
저장소 구조	분산 구조	분산 구조	중앙/분산 구조
질의 형식	지역 질의	글로벌 질의	글로벌 질의

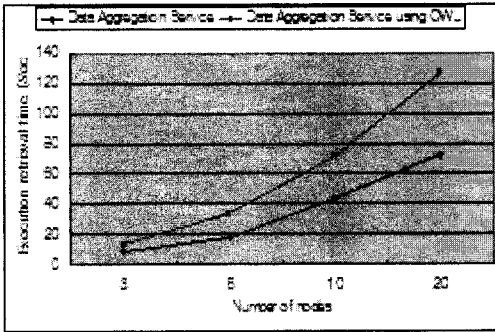


그림 8. 의미 조건 포함 질의 처리 성능
Figure 8. Query Process Performance with Semantic condition

VI. 결론 및 향후연구

본 논문은 데이터 허브를 이용하여 Legacy DB들의 통합 및 상호운용이 가능한 프레임워크를 제안하였다. 제안한 프레임워크는 데이터 상호운용에서 나타나는 이질적인 문제를 XMDR이 해결하고, 데이터의 교환, 공유, 검색은 데이터허브와 미디어데이터에서 그리고 상호운용에 의해 수집된 데이터를 저장하는 역할은 Proxy DB에서 하도록 구성되었다. 따라서 본 프레임워크는 분산된 Legacy DB간의 협업 환경 구성에 적합하며, 실시간 기업환경에서 빠른 정보 전달과 업무 지원 환경에 적절한 프레임워크이다. 이후는 워크플로우 시스템간의 데이터 상호운용이 되도록 확장하여야 하며, 기업환경에서 다양한 서비스를 제공하기 위한 서비스 계층이 확장되도록 데이터를 지식관리 수준까지 향상시키는 연구가 필요하다.

참고문헌

[1] Andrew B. Whinston and Varghese S. Jacob. "Electronic Commerce." Information Technology & Management 1.,2000.
[2] Fred A. Cummins, Enterprise Integration: An Architecture for Enterprise Application and Systems Integration, Wiley; 1st edition, February 1, 2002.

[3] Kevin D. Keck and John L. McCarthy, "XMDR: Proposed Prototype Architecture Version 1.01", <http://www.xmdr.org>, February 3, 2005
[4] W.Kim and J.Seo, "Classifying schematic and data heterogeneity in multi-database systems," IEEE Computer, Vol.24, No.12, pp.12-18, 1991.
[5] 문석재, 정계동, 최영근, "분산 데이터 상호운용을 위한 SQL/XMDR 메시지 기반의 Wrapper를 이용한 데이터 허브 시스템", 한국해양정보통신학회 제11권 11호 p2047~2058, 2007.11
[6] 문석재, 이수연, 최영근, "브리지 XMDR 기반의 메시지 교환 방식을 이용한 협업 시스템 설계", 한국해양정보통신학회 제11권 1호 p56~p65, 2007.1
[7] 웹 환경에서 데이터 상호운용을 위한 XMDR 기반의 검색 시스템 설계", 한국해양정보통신학회 제10권 12호 p2212~p2220, 2007.12
[8] 정계동, 황치근, 최영근, "분산 환경에서 XMDR을 이용한 예약 정보 시스템", 한국해양정보통신학회 논문지 Vol.11 No.10 pp.1957-1967, 2007
[9] 황치근, 정계동, 최영근, "지식 공유 기반의 XMDR을 이용한 적응형 검색 시스템 설계", 한국통신학회 논문지 제31권 제8호 p.716-729, 2006.
[10] 정계동, 황치근, 최영근, "분산 환경에서 XMDR 기반의 멀티데이터 베이스 상호운영 모델 설계", 한국해양정보통신학회논문지 Vol.11 No.9 pp.1771-1780, 2007

저자소개



문석재 (Seok-Jae Moon)

2002년 방송통신대학교 전자계산학 졸업
2004년 광운대학교 컴퓨터소프트웨어학과 석사
2004년~2005년 필컴정보시스템 주임연구원
2006년~ 현재 광운대학교 컴퓨터과학과 박사과정
※ 관심분야: XMDR, 데이터 그리드, 상호운용성



정계동(Gye-Dong Jung)

1985년 광운대학교 전자계산학 졸업
1992년 광운대학교 산업정보학석사
2000년 광운대학교 컴퓨터과학박사
1993년~2004년 광운대학교 정보과
학원 교수

2005년~ 현재 광운대학교 교양학부 교수

※관심분야: XML 분산시스템, 분산 컴퓨팅기술, 이동
에이전트



최영근(Young-Keun Choi)

1980년 서울대학교 수학교육과
이학사
1982년 서울대학교 계산통계학과
이학석사

1989년 서울대학교 계산통계학과 이학박사

1983년~ 현재 광운대학교 컴퓨터과학과 교수

1992년~2000년 광운대학교 전산정보원 원장

2002년~2005년 광운대학교 교무연구처장

※관심분야: 객체지향설계, 분산시스템, 이동에이전
트, 상호운용성