DOI: 10.3745/KIPSTD.2009.16D.5.701

협업을 위한 XMDR-DSM 시스템 설계에 관한 연구

문 석 재[†] · 정 계 동[††] · 최 영 근[†††]

요 약

서비스 기반의 전사적 데이터 통합은 기업 변화의 가속화에 따른 통합적인 데이터 관리가 필요하다. 기업 간 협업은 비즈니스 프로세스를 이용하여 분산된 데이터베이스들에 접근하여 이루어진다. 이러한 접근은 글로벌 쿼리 기반으로 수행되기 때문에 이기종간의 상호운용을 위해서 데이터 이질성, 스키마 이질성, 유효성 검증과 같은 문제가 선결되어야 한다. 따라서 협업은 동적이면서 신뢰성 있는 구조를 갖추어야 한다. 본 논문에서는 협업을 위한 XMDR-DSM(eXtended MetaData Registry-Data Service Mediator) 시스템을 제안한다. XMDR-DSM은 XMDR-DS와 XMDR-DQP, XMDR-DAI로 구성되어 글로벌 스키마와 로컬 스키마간의 매핑을 지원해주고, 데이터 접근 및 통합 서비스를 제공한다. 따라서 이기종 데이터베이스 상호간의 업무 지원이 가능하게 하고, 정보를 재활용 가능한 자산으로 확보할 수 있는 이점과 교환의 표준화를 제시할 수 있다. 그리고 워크플로우 기반의 비즈니스 프로세스 제공할 수 있으므로 일원화된 정보를 관리할 수 있게 된다. 따라서 정보 수명을 증대하고 비용절감을 할 수 있게 된다.

키워드 : 이기종 데이터베이스, 협업, 메타데이터, 비즈니스 프로세스, 상호운용

A Study on XMDR-DSM System Design for Cooperative

SeokJae Moon[†] · KyeDong Jung[††] · YoungKeun Choi[†††]

ABSTRACT

In the enterprises the data integration based on service requires integrated data management as the change in the environment of enterprises accelerates. Cooperation among enterprises is accomplished through accessing distributed database using business process. As this approach is performed based on the global query, problems such as data heterogeneity, schema heterogeneity, and verification of validity have to be solved in advance for the interoperability among the heterogeneous system. Thus, cooperation requires dynamic and reliable construction.

In this paper, we propose XMDR-DSM (eXtended MetaData Registry-Data Service Mediator) system for cooperation. XMDR-DSM, which is comprised of XMDR-DS, XMDR-DQP, and XMDR-DAI, supports the mapping between global schema and local schema and provides data access and integration service. Therefore, XMDR-DSM enables the mutual support of business operations among heterogeneous database. In addition, it can secure information as reusable asset and the standardization of interchange. Also it can manage unified information since it provides business process based on workflow; therefore, it will be able to increase the life span of information and reduce the cost.

Keywords : Heterogeneous DataBase, Cooperative, MetaData, Business Process, Interoperability

1. 서 론

기업의 다양한 통합 요구에 따라 급변하는 비즈니스 환경에서의 B2B 또는 기업 내부에서 프로세스 통합에 대한 요구가 증가하고 있다[1, 2]. 이런 상황 속에서 협업은 이기종 시스템간의 상호운용이 필수적이다. 그러나 애플리케이션 통합 문제에 따른 이기종 플랫폼 간 데이터 상호운용에 의한 협업 처리에 어려움이 발생한다. IT환경에서는 이기종 플랫폼에 따른 기업 간 협업 문제를 선결해야 할 과제가 많다[3]. 그 중에 가장 시급한 부분이 표준화이다. 표준화를 통해 오늘날 비즈니스 업무들은 오래 동안 직면하고 있는 통합 문제들을 해결할 수 있게 되었다[4]. 현재는 이기종 통합에서 웹 서비스의 사용은 사용자들이 이미 알고 있는 서비스만을 이용하거나, 키워드 기반의 검색엔진 또는 웹 서비스 레지스트리[5] 탐색에 의해 적합한 서비스들을 검색하고 있다. 또한 검색된 서비스들의 조합은 주로 사용자 판단에

† 준 회 원 : 광운대학교 컴퓨터과학과 박사과정
†† 정 회 원 : 광운대학교 교양학부 교수
††† 정 회 원 : 광운대학교 컴퓨터과학과 교수
논문접수 : 2009년 7월 20일
수 정 일 : 1차 2009년 8월 17일, 2차 2009년 9월 1일
심사완료 : 2009년 9월 2일

의하여 수작업으로 이루어지므로 이는 매우 불편한 작업이 될 수 있다[6, 7].

기업에서 협업은 서비스 기반의 전사적 데이터 통합 프로세스 중심이며, 핵심은 비즈니스 프로세스이다. 비즈니스 프로세스(business process)란 기업이 타 기업과 개인 고객을 위한 모든 파트너에게 가치(서비스, 제품)를 제공하기 위해 순차적이거나 동시적으로 발생할 수 있는 모든 연관된 업무들의 모음을 의미한다. 이러한 비즈니스 프로세스 기반의 업무들은 워크플로우를 이용하여 자동적으로 수행할 수 있어야 한다[2]. 워크플로우는 기업의 비즈니스 프로세스를 GUI도구를 통해 모델링 과정을 거쳐 설계를 하고, 설계된 프로세스를 실제로 운영하면서 다양한 프로세스의 성능을 모니터링하고 관리하는 시스템을 의미한다. 현재 수많은 기업들이 비즈니스 프로세스를 자동생성 할 수 있는 BPM (business process management) 솔루션 시장에 합류하고 있다. 더욱이 BPM 솔루션을 이용한 데이터 기반의 전사적 데이터 통합 서비스는 기업 변화의 속도가 가속화됨에 따라 데이터 관리에 직접 영향을 미치는 인프라에 대한 빠른 대응력을 향상시켜야한다는 요구가 높아지고 있다. 이에, 기업들은 데이터 서비스 중심의 협업 시스템 구축에서 비즈니스 프로세스를 이용하여 상호운용 서비스가 지원이 되도록 하려고 한다.

본 논문에서는 비즈니스 프로세스를 통해 상호간의 협업 지원이 가능한 시스템 구축을 위한 미들웨어인 XMDR-DSM을 제안한다. XMDR-DSM(eXtended MataData Registry-Data Service Mediator)는 비즈니스 프로세스의 처리 과정이 기술된 워크플로우를 이용한 협업을 효율적으로 지원할 수 있게 로컬 데이터베이스간의 데이터 상호운용상의 브리지적인 역할을 담당하고, 협업 프로세스에 따른 데이터 페더레이션(Data Federation), 데이터 마이그레이션(Data Migration), 데이터 통합(Data Integration)이 가능하게 된다. 비즈니스 프로세스는 로컬 데이터베이스간의 메타데이터 스키마 표준화 과정을 거친 글로벌 스키마를 참조하여 정의된 업무들로써 즉, 글로벌 쿼리 기반의 프로세스이다[7]. XMDR-DSM은 협업에 의해 미리 정의된 비즈니스 프로세스를 웹 서비스에 등록하여 로컬 데이터베이스간의 데이터 서비스 중심의 업무 처리를 할 수 있으며, 웹 서비스 기반의 워크플로우를 정의하여 협업 처리를 할 수 있게 한다. 그리고 협업에 필요한 데이터 상호운용에서 발생하는 여러 가지 이질성들에 대한 문제는 XMDR을 이용하여 해결한다[8]. 또한 XMDR 기반의 DSM은 글로벌 스키마와 로컬 스키마간의 스키마 변환 및 데이터 구조, 형식, 의미, 표현 정보를 제공함으로써 로컬 시스템간의 데이터 공유 및 교환을 할 수 있도록 지원한다[8-11].

XMDR-DSM은 XMDR-DQP(Distributed Query Processor)와 XMDR-DAI(Data Access and Integration)으로 구성된다. XMDR-DQP는 XMDR을 참조하여 데이터 통합 서비스 중심의 분산 데이터를 액세스할 수 있도록 글로벌 쿼리와 로컬 쿼리간의 변환 기능을 제공한다. 여기서 XMDR-DQP는 XMDR-DAI와 연계되어 사용자가 로컬 데이터베이스에 액세스 및 트랜잭션을 하기 위한 쿼리를 로컬 쿼리로 재정의하는 기능을 제공한다. 그리고 사용자에게 일관성 있는 글로벌 스키마를 보여줌으로써 분산된 로컬 데이터베이스에 별도의 스키마 지식이 없이도 워크플로우 기반의 비즈니스 프로세스를 정의할 수 있게 한다.

따라서 본 논문의 목적은 로컬 데이터베이스 간 협업을 통해 비즈니스 프로세스를 처리할 수 있게 환경을 구축하고, 웹 서비스를 이용한 이질적인 데이터들 간의 상호운용이 가능하게 하는 효율적인 서비스 지원하도록 하기 위한 것이다. 또한 프로세스 재정의에 의한 처리, 새롭게 정의되는 프로세스를 사용자가 작성하여 처리할 수 있게 지원한다. 이는 분산 데이터의 일관성을 유지하면서 공유 및 교환되는 데이터의 활용을 지원하여 가용성을 향상시킬 수 있는 협업을 위한 프레임워크를 구축할 수 있다. 이러한 구축은 정보를 재활용 가능한 자산으로 확보할 수 있는 이점과 정보 제공 및 교환의 표준화를 제시할 수 있다. 그리고 비즈니스 프로세스를 통해 표준화된 정보를 관리할 수 있게 된다. 따라서 정보 수명을 증대하고 비용절감을 할 수 있게 된다.

본 논문의 구성은 다음과 같다. 2장은 상호운용시스템, XMDR, 워크플로우 기반의 비즈니스 프로세스에 대한 관련 연구를 기술하고, 3장은 XMDR-DSM을 이용한 기반시스템 프레임워크 구성을 기술하고, 4장은 쿼리 처리 메커니즘을 기술한다. 5장은 Proxy-DB를 이용한 데이터 이주 기법을 기술하고, 6장은 적용사례 및 비교평가를 7장은 결론 및 향후연구에 대해 기술한다.

2. 관련 연구

2.1 상호운용 시스템

기업들은 분산 환경에서 데이터베이스 또는 레거시간의 협업 시스템을 구축하려는 시도를 하고 있다. 이런 협업 시스템은 데이터 서비스가 목적인 협업에서 분산된 데이터베이스나 레거시에게 균일한 인터페이스를 제공하고 데이터 자원의 대한 이동, 접근, 업데이트가 지원될 수 있어야 한다. 특히 데이터 이질성을 해결하고자 하는 노력은 많은 연구에서 찾아볼 수 있다. 이러한 분산 데이터의 상호운용에 관한 연구는 최근의 로컬 시스템에서 운영되는 데이터베이스를 상호 연결해야하는 필요성에 따라 협업을 형성하고, 상호운용 기술을 기반으로 한 시스템 환경 개선에 많은 연구가 진행 중에 있다[2]. 이기종 협업 시스템의 데이터의 상호운용을 위한 대표적인 시스템으로 GRelC(Grid Relational Catalog) 프로젝트에서 제안한 Grid-DBMS[12]는 그리드 환경에서 동적으로 데이터 관리를 위한 프레임워크이다. 또 다른 시스템으로 실시간 기업환경에서 정보통합관리를 위한 OGSA-WebDB[13]는 GDS(Grid Data Service)를 이용하여 실시간으로 데이터를 통합관리하기 위한 시스템이다. Grid-DBMS는 논리적으로 구성되는 가상 데이터 공간에서 동적으로 그리드 데이터베이스를 관리하고 데이터를 상호운용하

기 위하여 제안한 프레임워크이다. Grid-DBMS에서 정의한 그리드 데이터베이스는 논리적으로 하나 이상으로 구성되는 이종의 데이터베이스이며, 레거시 시스템에서 독립적으로 운영되어 상호 연동될 수 있는 데이터베이스들의 집합이다. 이러한 Grid-DBMS 프로젝트에서 적용한 그리드 미들웨어는 Globus Toolkit[14]을 사용하였으며, 데이터 교환은 Grid-FTP[15]를 사용하였다. 데이터 교환을 위한 작업으로 데이터 단편화와 재할당, 그리고 데이터 복사 작업을 지원한다.

OSGA(Open Grid Services Architecuture)를 기반으로 한 OGSA-WebDB[13]는 기존에 운영되던 레거시 시스템들이 실시간 협력을 위한 중앙 집중 구조로서 데이터를 통합 운영하기 위하여 제안한 프레임워크이다. GDS는 레거시 시스템들의 실시간 변화 환경을 지원하며, 데이터를 통합 운영하기 위한 허브 유형 시스템이다. OGSA-WebDB는 데이터 자원들을 지원하기 위한 미들웨어로서, 이종 데이터베이스에 대한 접근과 분산 및 통합처리를 위한 획일적인 서비스 인터페이스를 제공하는 것이다.

따라서 OGSA-WebDB는 데이터 통합 및 분산 질의와 같은 고차원 서비스들에서 사용될 수 있는 기본적인 오퍼레이션들을 제공한다. 이와 같이 OGSA-WebDB가 제공하는 기능은 주로 데이터간의 상호운용을 초점으로 하여 연구되어지고 있다.

본 논문에서는 협업 환경에서 비즈니스 프로세스를 지원하기 위한 XMDR-DSM을 제안하며, XMDR을 이용하여 이기종간의 비즈니스 프로세스를 수행하는 과정에서 발생하는 상호운용 이질성을 극복하여 데이터를 통합함으로써 정보의 재사용 및 정보 수명이 증대되고 비용절감을 할 수 있게 한다. 또한 기존 정의된 비즈니스 프로세스들의 유기적인 연계 과정을 보다 용이하게 한다.

2.2 XMDR

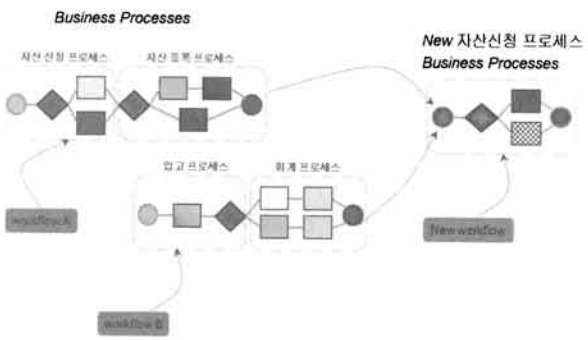
XMDR은 메타데이터의 데이터 요소, 용어 및 구조적 개념을 의미적으로 저장하고 검색하기 위한 기술로 XMDR은 ISO/IEC 11179[8, 16]에서 제안한 정보 공유 및 교환을 위한 표준으로 현재 많은 프로젝트가 진행 중에 있다. 특히 ISO/IEC 11178-3에서 공유 데이터의 관리를 위한 메타 모델, 기본 속성이 제시되었는데, 메타 모델은 의미적인 내용과 그리드 환경하의 사용자들이 정보처리 시스템간의 공유되는 데이터 요소의 구문을 위한 표준과 안내를 제공하고 있다. 본 논문에서의 DSM은 XMDR을 이용하여 명확한 요구사항을 만족하도록 구성된 것으로써, 효율적인 데이터 서비스를 위한 데이터 상호운용을 수행할 수 있다. 그리고 그리드상의 분산 데이터의 자율성과 독립성을 보장할 수 있고, 기본 데이터베이스들의 메타데이터 모델을 설계하여 XMDR의 구성요소 및 정의를 기술한 것이다.

- *Meta-Semantic Ontology(MSO):* MSO는 데이터베이스들의 메타데이터 스키마 정보를 시소러스화한 것으로, 표준으로 지정한 메타데이터 스키마에 매핑하여 메타데이터간의 관계성과 이질성을 해결할 목적으로 정의한 것이다.
- *Instance-Semantic Ontology(InSO):* InSO는 실제 인스턴스 값(value)과 분산 인스턴스 값(value)들 간의 연관 관계성을 시소러스화 한 것으로, 인스턴스 값 사이에 의미성, 유사성을 고려하여 정의한 것이다.
- *Meta-Location(MLoc):* MLoc는 MSO와 연계하여 로컬 데이터베이스들의 물리적인 정보인 위치, 접근, 권한 등을 등록한 것으로, 데이터 접근 및 통합에서의 상호운용에 필요한 데이터 이주과정이나 질의 프로시저 메시지가 해당 위치에 전달될 수 있게 하는 목적으로 정의한 것이다[9-11].

2.3 워크플로우 기반의 비즈니스 프로세스

기업 내에는 여러 가지 비즈니스 단위에 프로세스가 존재한다. 이러한 프로세스들은 특정한 작업 흐름을 갖는 단위들로 구성된다. 워크플로우는 비즈니스 프로세스의 자동화를 의미하며, 워크플로우 관리 시스템(WfMS)은 소프트웨어를 이용하여 워크플로우의 수행을 정의, 생성, 관리하는 시스템을 의미한다[13]. 대부분의 워크플로우 시스템은 업무 흐름의 자동화, 정보 및 문서 전달의 전자화 그리고 일관적 데이터 접근, 제어를 통해 비즈니스 프로세스의 개선, 통제, 관리, 공동 작업을 지원한다. 그리고 WfMS는 비즈니스 프로세스의 개선 측면에서 BPR(business process reengineering)이 한 가지 방법으로 생각되기도 한다. WfMS는 비즈니스 프로세스의 자동화를 의미하며 문서, 정보, 업무 등의 특정 사용자에서 다른 사용자에게로 사전에 정의된 규칙에 의하여 전달된다. 이러한 WfMS의 적용 대상은 기업 전체이므로 기업 내의 모든 업무와 자원을 프로세스 관점에서 통합하여 관리하게 된다. WfMS는 기업의 업무 흐름을 자동으로 처리함으로써 이에 따른 비용과 시간을 절약할 수 있는 솔루션이다. 국내외 대부분의 연구들이 워크플로우의 비즈니스 프로세스 정의 및 인스턴스 생성, 실행에 중점을 두고 진행되고 있으며, 워크플로우 상호운용성에 관해서는 워크플로우 표준에서 정의한 상호운용성 명세를 그대로 구현해서 사용하는 수준의 연구가 진행되고 있다.

본 논문에서의 비즈니스 프로세스는 (그림 1)과 같이 각각의 프로세스를 XMDR에 정의된 글로벌 스키마 기반으



(그림 1) 워크플로우 기반의 비즈니스 프로세스

로 작성한 것이고, 워크플로우는 협업된 로컬 데이터베이스마다 순차적으로 처리해야 될 수행 과정을 기술한 것이다. 예를 들어 "자산 신청 처리 업무를 작성하라"와 같은 요구에 대해 비즈니스 프로세스인 수행 순서를 (그림 1)과 같이 작성한 것이다.

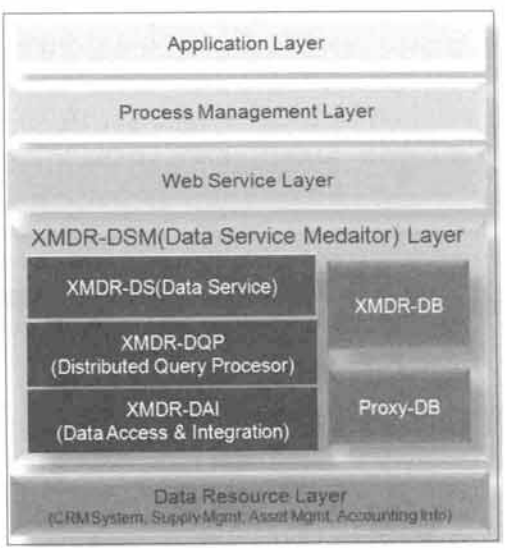
이는 자산신청, 자산등록, 입고처리, 회계처리에 대한 새로운 프로세스 워크플로우를 설계한 것으로 각 프로세스들은 본 논문에서 제안한 XMDR-DSM을 통해 비즈니스 레벨의 협업을 처리 할 수 있다.

3. 협업을 위한 XMDR-DSM 시스템 설계

3.1 협업을 위한 시스템 전체적인 구성도

본 논문에서의 XMDR-DSM은 전사적 데이터 통합 서비스를 제공함으로써 협업을 효율적으로 처리할 수 있도록 설계하였다. 이는 글로벌 스키마 기반으로 작성된 비즈니스 프로세스가 효과적으로 분산된 이종 데이터에 접근하여 수행할 수 있도록 구현된 것이다. 글로벌 스키마 기반의 비즈니스 프로세스는 이종 데이터들로부터 정보 검색, 삽입, 삭제, 변경이 발생할 때 스키마간의 의미충돌을 감지하고 해결하는 일이 주기능이다. 또한 이종 데이터 상호운용을 수행할 때 데이터에 대한 이질성, 유효성 검증과 같은 문제를 해결하도록 설계되었다. (그림 2)와 같이 협업을 위한 XMDR-DSM 시스템의 구성요소는 애플리케이션, 프로세스 관리, 웹 서비스, XMDR-DSM(XMDR-DS, XMDR-DQP, XMDR-DAI), 데이터 리소스 계층으로 구성된다. 다음은 계층별 주요 역할은 다음과 같다.

(1) 애플리케이션 계층(Application Layer)은 사용자가 직접적으로 협업 프로세스를 정의하는 역할을 한다. 이는 사용자가 비즈니스 프로세스를 정의하는 GUI도구를 이용해서



(그림 2) 협업을 위한 XMDR-DSM 시스템 구성

프로세스 템플릿 레포지토리에 정보와 XMDR-DB에서 제공하는 글로벌 스키마를 기반으로 비즈니스 프로세스를 정의한다.

(2) 프로세스 관리 계층(Process Management Layer)은 미리 정의된 협업 프로세스 별로 템플릿을 구성하여 관리하고, 이것을 사용자에게 제공하는 영역이다. 이는 통합 업무 인터페이스를 통해 분산된 이기종간의 협업할 비즈니스 프로세스를 템플릿 형태로 제공함으로써 사용자는 원하는 업무를 쉽게 정의할 수 있게 된다.

(3) 웹 서비스 계층(Web Service Layer)은 협업에 맞게 정의된 비즈니스 프로세스의 정보가 UDDI에 등록되어져 있다. 사용자는 협업에 필요한 프로세스를 웹 서비스를 통해 검색하여 WSDL 기반의 비즈니스 프로세스를 정의해서 처리 할 수 있게 제공한다.

(4) XMDR-DSM 계층(XMDR-DSM Layer)은 XMDR-DS와 XMDR-DQP, XMDR-DAI로 구성되어 글로벌 스키마와 로컬 스키마간의 매핑을 지원해주고, 데이터 접근 및 통합 서비스를 제공한다.

- *XMDR-DS*(Data Service)는 글로벌 스키마 기반의 프로세스를 XMDR-DQP에게 전달하거나 Proxy-DB에 저장된 프로세스 수행 결과를 애플리케이션에 전달하기 위한 역할을 담당한다.
- *XMDR-DQP*(Distributed Query Processor)는 글로벌 스키마 기반의 프로세스의 신택스(syntax)를 파싱(parsing)하여 해당 로컬 데이터베이스에 적합한 로컬 쿼리로 변환한다. 만약 사용자가 새롭게 비즈니스 프로세스를 정의하여 생성된 것은 프로세스 템플릿 레포지토리에 등록하고, 프로세스 실행 후 결과를 수집하기 위한 table은 Proxy-DB에 생성한다. 하지만 사용자가 업무상 일시성인 프로세스 쿼리를 직접 작성하여 수행할 경우에는 Proxy-DB에 임시 table이 생성되어 협업 상의 데이터 상호운용을 원활하게 해주는 역할도 한다. 글로벌 스키마 기반의 프로세스 구문을 파싱하여 로컬 데이터베이스에 맞는 로컬 쿼리로 매칭, 변환하는 기법은 본 논문의 4장에서 다루고 있다.
- *XMDR-DAI*(Data Access and Integration)는 데이터 수집 및 처리 에이전트의 일종이다. 마치 자신의 시스템 내에 있는 로컬 에이전트처럼 취급할 수 있다. 로컬 스키마 기반으로 변환된 프로세스를 각 로컬 데이터베이스에서 실행하여 프로세스 수행 결과를 다시 Proxy-DB에 보낸다. 다시 말해, XMDR-DQP에서 변환하여 보낸 로컬 쿼리문을 해당 로컬 데이터베이스에 요청하여 데이터 수집, 트랜잭션, 이주 과정을 처리할 수 있게 해준다.
- *XMDR-DB*는 협업된 로컬데이터베이스의 메타데이터 스키마와 글로벌 스키마의 매핑정보를 등록한 것이다.
- *Proxy-DB*는 비즈니스 프로세스 워크플로우 수행 과정을 통해 로컬 데이터베이스간의 데이터 이주에 따라 실행, 처리되는 데이터 보관소이다. 자세한 처리 과정

은 5장에서 설명하였다.

(5) 데이터 리소스 계층(DRs:Data Resource Layer)은 기존에 구축되어 운영되고 있는 레거시 시스템 즉, 데이터베이스 관리 시스템이 존재하는 영역이다.

3.2 협업을 위한 XMDR-DSM 시스템 동작 과정

이 절은 XMDR-DSM 이용한 협업 시스템에서의 전체적인 동작 흐름을 나타낸 것으로 (그림 3)과 같다. 동작 흐름에 대한 설명은 통합 유형 자산 관리 시스템에서 (그림 1)의 자산신청 프로세스를 수행하는 시나리오로 설명한다. 자산신청 프로세스는 조회, 신청, 등록 쿼리문으로 구성된다.

(그림 3)에 도식화 한 것처럼 (1)의 과정을 통해 Client/Application에게 웹 서비스에 등록되어 있는 자산신청 프로세스 리스트를 넘겨준다. 이는 (그림 15)의 ①에서처럼 자산신청에 필요한 프로세스가 미리 정의된 것으로 사용자는 프로세스의 수행 순서를 정의한다. 또한 자산신청 프로세스가 웹 서비스에 등록되어 있다면 (그림 3)에 나머지 과정을 수행하지 않고 바로 XMDR-DAI를 통해 프로세스를 처리할 수 있다. (2)의 과정은 웹 서비스에 자산신청 프로세스가 등록되어 있지 않은 경우, 프로세스 템플릿 레포지토리에서 새롭게 정의할 템플릿을 제공한다. 이렇게 제공된 프로세스 템플릿 리스트와 XMDR-DB에서 (3)의 과정을 통해 넘겨받은 글로벌 스키마 정보를 가지고 사용자는 자산신청 프로세스의 내부 쿼리를 (그림 15)와 같이 정의한다. 사용자에 의해 글로벌 스키마 자산신청정의된 쿼리는 (4)의 과정에서 XMDR-DS에게 넘겨주게 된다. (5)의 과정은 과정에서 XMD가 새롭게 정의한 자산신청 프로세스를 템플릿 레포지토리에 등록하고, 프로세스 수행 후 의해 글보관할 스키마 자산신청정미리 table을 생성한다(그림 16). 그 후 (6)의 과정을 통해 과정에서 XMD는 XMDR-DB를 통해 받은 로컬 스키마 정보와 새롭게 정의된 자산신청 프로세스를 (그림 10)에 같르핑하는 과정을 통해 각쿼리를 (그림 15)와 수행될 게 된다. 는 자산신청 프로다. 이렇게 프로세 된다. 는 자의 자산신청 프로세스는 (7)의 과정을 통해 웹 서비스에 등록되는 동시에 (8)의 과정을 통해 DRs의 쿼리XMDR-DAI에 프로세스가 수행되게 된다. 자산신청 프로세스가 쿼리를 데이터베이스에서 수행되어 온 결과는 (9)의 과정에서 Proxy-DB에 삽입되고, (10),(11) 과정에서 XMDR-DS를 통해 사용자에게 결과를 보여주게 된다.

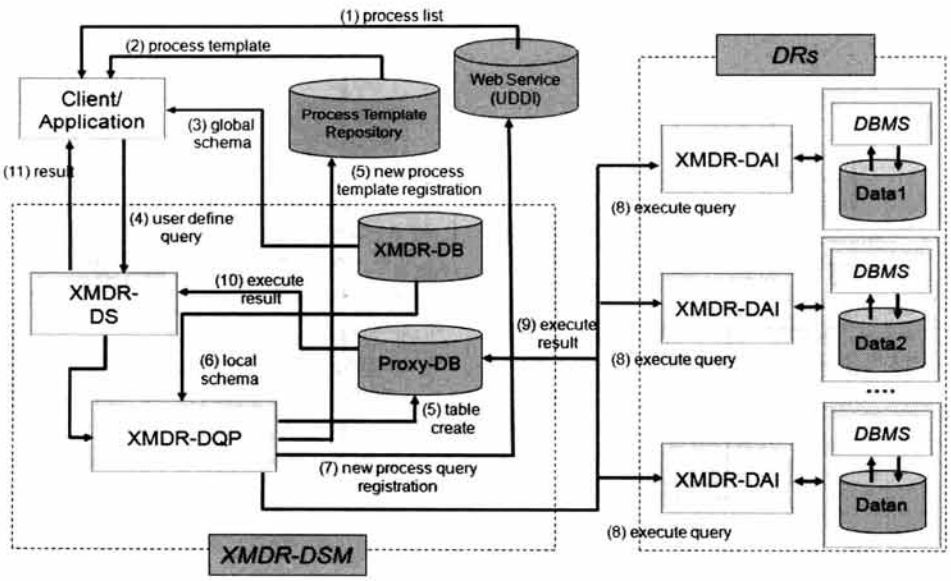
3.3 XMDR-DQP 구성

본 논문에서 제안한 XMDR-DSM의 XMDR-DQP는 (그림 4)에서처럼 5개의 구성요소로 이루어진다. 이는 글로벌 쿼리문을 로컬 쿼리문으로 변환하여 각 로컬 데이터베이스에 적합하도록 설계되었다. 이는 여러 다른 협업 데이터모델로 표현된 로컬 스키마를 하나의 단일한 글로벌 스키마로 전환시킴으로써 일관성을 보장한다. XMDR-DQP의 구성요소에 대한 역할은 다음과 같다.

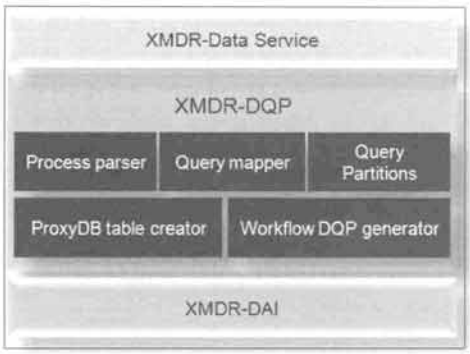
*(1) Process parser*는 사용자가 작성한 글로벌 스키마 기반의 비즈니스 프로세스를 파싱하여 글로벌 메타데이터 스키마 정보를 추출하는 기능을 한다. 글로벌 스키마 정보를 이용하여 Proxy-DB table creator에게 전달하여 table을 생성하는데 필요한 정보를 제공한다.

*(2) Query mapper*는 XMDR-DB에 등록된 메타데이터 스키마 정보를 이용하여 분석된 글로벌 스키마와 로컬 스키마에 간의 매핑을 해주는 기능을 한다. 이는 로컬 스키마로 변환하는 매핑 연관 정보를 제공함으로써 각 로컬 데이터베이스에 맞는 쿼리문을 작성하는데 있다.

*(3) Query Partition*은 로컬 스키마 기반으로 변환된 쿼리문으로 각 로컬 데이터베이스에 맞게 분할하는 기능을 한



(그림 3) 협업 시스템 전체 흐름



(그림 4) XMDR-DQP 구성요소

다. 이는 비즈니스 프로세스 내부에서 정의된 수행순서를 정하는 워크플로우를 생성하는데 용이해진다.

*(4) Workflow DQP generator*는 DRs(Data Resource Layer)에 수행될 로컬 쿼리문들의 순서를 정의한 WSDL 기반의 워크플로우를 생성하는 기능과 Web Service에 새롭게 정의된 워크플로우 기반의 비즈니스 프로세스를 등록하는 기능을 담당한다.

*(5) ProxyDB table creator*는 파싱된 메타데이터 스키마 정보를 이용하여 Proxy-DB에 table을 생성하는 기능을 한다. 이는 프로세스가 실행된 후에 그 결과 데이터를 저장하는 임시 테이블이다.

3.4 XMDR-DQP 동작 과정

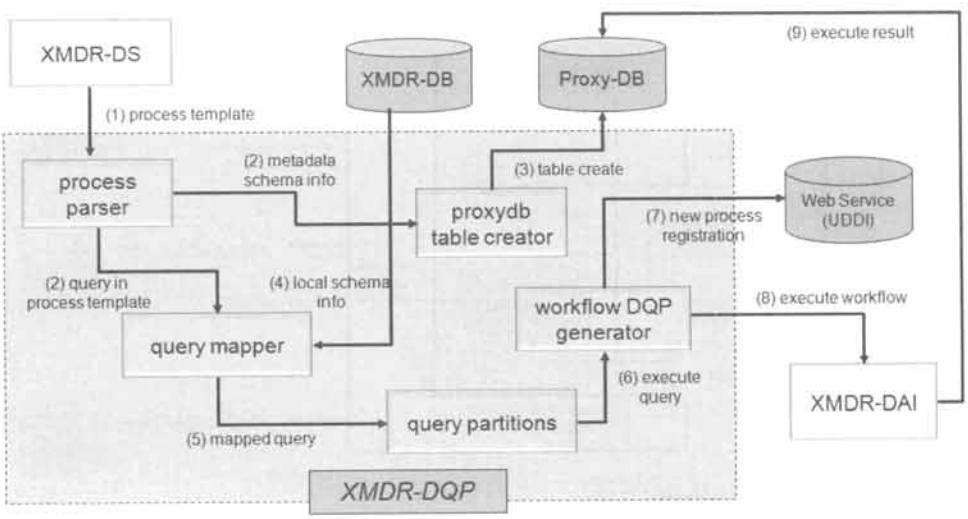
이 절은 XMDR-DQP의 전반적인 동작 흐름을 나타낸 것으로 (그림 5)와 같다. XMDR-DS에서는 3.2절에서 사용자가 새롭게 정의한 자산신청 프로세스를 (1)의 과정을 통해 process parser로 넘겨준다. process parser는 자산신청 프로세스를 파싱하여 메타데이터 스키마 정보와 내부 쿼리 정보를 각각 분리한다. 이렇게 분리된 정보에서 스키마 정보는 (2)의 과정을 통해 proxydb table creator에게 전달하여 (그림 16)과 같이 Proxy-DB에 table을 생성한다. 나머지 내부 쿼리 정보는 글로벌 스키마 기반이기 때문에 (그림 10)에 매핑 과정을 통해 로컬 스키마 기반의 쿼리로 변환되고 동작과정은 (그림 18,19,20)과 같다. 이렇게 변환된 로컬 쿼리는 (5)의 과정을 통해 query partition으로 전달되고, 실제 로컬 데이터베이스 맞게 쿼리가 분할된다. 이는 다시 workflow DQP generator에 전달되어 분할된 로컬 쿼리문의 수행 순서를 WSDL기반의 워크플로우로 생성하고, 새롭게 정의된 프로세스 워크플로우는 (7)의 과정을 통해 웹 서비스(UDDI)에 등록한다. 이 (7)의 과정은 (그림 15)에서처럼 사용자에게 웹 서비스에 등록되어 있는 프로세스 리스트를 제공하는데 이용된다. 이런 작업이 완료되면 (8)의 과정을 통해 최종적으로 DRs에 있는 각 로컬 데이터베이스에 데이터 접근 및 처리를 담당하는 에이전트인 XMDR-DAI에 전달된다. 이렇게 전달된 워크플로우는 XMDR-DAI에 의해서 정의된 수행 순서대로 쿼리문을 실행하고, 그 결과를 다시 (9)의 과정을 통해 Proxy-DB에 저장한다. 이때 Proxy-DB table creator에 의해 미리 생성된 table에 저장되어 진다. 이는 사용자가 프로세스 수행에 따른 결과 데이터를 직접적으로 확인할 수 있으면, 필요에 따라 가동되어 다시 사용되기도 한다.

3.5 XMDR-DAI 구성

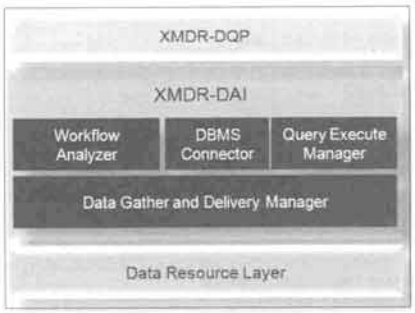
본 논문에서 제안한 XMDR-DSM의 XMDR-DAI는 (그림 6)처럼 4개의 구성요소로 이루어진다. 이는 XMDR-DQP에서 전달한 로컬 스키마 기반의 워크플로우 문서를 파싱하여 각 로컬 데이터베이스에 컨넥션하여 쿼리문을 실행하고, 실행 후 결과를 수집하여 Proxy-DB에 저장하도록 설계되었다. XMDR-DAI의 구성요소에 대한 설명은 다음과 같다.

*(1) Workflow Analyzer*는 XMDR-DQP에서 전달된 로컬 스키마 기반의 워크플로우 문서를 파싱하여

분석하는 기능을 담당한다. 이로 인해 각 쿼리문이 워크플로우에 기술된 순서에 의해 해당 데이터베이스에 접근할 수 있다.



(그림 5) XMDR-DQP 동작 흐름



(그림 6) XMDR-DAI 구성요소

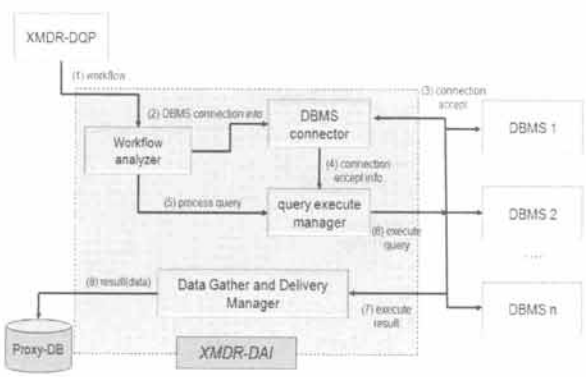
*(2) DBMS Connector*는 워크플로우 분석기에서 분석된 정보에서 각 로컬 데이터베이스의 컨넥션 정보를 이용하여 각 DRs에 접근 인증권을 얻어 실제 실행되는 쿼리문에 적용할 수 있게 쿼리 실행 관리자에게 전달하는 기능을 담당한다.

*(3) Query Execute Manager*는 워크플로우 분석기에서 분석된 쿼리와 DRs의 접근 인증 정보를 가지고 실행 가능한 쿼리문을 생성 및 수행하는 기능을 한다.

*(4) Data Gather and Delivery Manager*는 각 로컬 데이터베이스마다 쿼리가 실행된 결과를 각각 수집하여 Proxy-DB에 전달하며 저장하는 기능을 담당한다. 이때 저장 기술은 XML 문서 기반의 데이터를 삽입하는 BulkLoad 기법을 이용한다.

3.6 XMDR-DAI 동작 과정

이 절은 XMDR-DAI의 전반적인 동작 흐름을 나타낸 것으로 (그림 7)과 같다. (1)은 (그림 5)의 (8)에 해당하는 과정을 통해 전달된 워크플로우다. (2)의 과정은 Workflow Analyzer를 통해 로컬 데이터베이스에 위치, 접근 정보를 DBMS Connector에 넘겨준다. DBMS Connector는 (3)의 과정을 통해 각 로컬 데이터베이스에 접근하여 인증권을 획득한 후, 획득한 권한을 다시 (4)의 과정을 통해 query execute manager에서 넘겨주고, (5)의 과정을 통해 워크플로우 분석기에서 넘어온 쿼리와 함께 실제 로컬 데이터베이스에서 실행 가능한 쿼리문으로 변경한 후에 (6)의 과정을 통해 실행하게 된다. 이렇게 실행된 후 결과 데이터들은 (7)의 과정을 통해 Data Gather and Delivery Manager에 의해 수집되어 Proxy-DB의 해당 table에 (8)의 과정을 통해서 삽입된다.



(그림 7) XMDR-DAI 동작 흐름

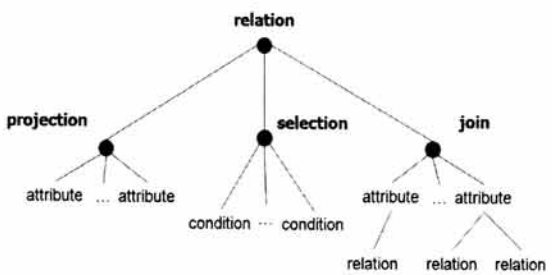
4. 쿼리 처리 기법

4.1 쿼리 파싱과 변환

본 논문에서 제안한 XMDR-DQP를 이용한 글로벌 스키마 기반의 쿼리를 로컬 스키마 쿼리로 변환하는 처리 방식은 비즈니스 프로세스가 사용자가 로컬 데이터베이스에 저장되어 있는 데이터에 접근하기 위해 반드시 XMDR에 등록되어 있는 글로벌 스키마를 통해서만이 접근이 가능하므로 강결합된(tightly-coupled) 연방 시스템 접근 방법과 유사하다고 할 수 있겠다. 따라서 글로벌 스키마 기반의 협업 프로세스의 처리와 쿼리에 대한 응답 서비스는 XMDR-DQP에서 지원하는 서비스를 통해 이루어진다. 이와 같은 방법의 장점은 로컬 시스템을 위해 개발된 응용프로그램에 대해 전혀 영향을 미치지 않으면서 통합된 뷰어를 제공하기 때문에 사용자는 이들 정보의 내용에 대한 사전 지식이 없어도 이질적인 데이터에 용이하게 접근할 수 있다.

XMDR-DQP를 이용한 협업 프로세스 기반의 글로벌 쿼리 처리 방식은 4단계를 거쳐서 수행하게 된다. 첫째, 쿼리 파싱과 분해 단계는 글로벌 쿼리를 분석하고 쿼리 그래프를 생성한다(그림 8). (그림 8)의 쿼리 그래프는 관계형 릴레이션(relation), 연산자(프로젝션(projection), 셀렉션(selection), 조인(join)), 속성(attribute), 조건문(conditions) 등의 다수의 노드로 이루어져 있는 트리 구조를 가지고 있다. 둘째, 쿼리 변환 단계는 글로벌 쿼리의 테이블 이름과 속성 이름을 해당 로컬 데이터베이스에 맞는 테이블 이름과 속성 이름으로 변환하는 작업을 한다. 이 단계에서 XMDR-DQP는 글로벌 쿼리의 조건문(SQL에서의 where 구문)의 의미를 수정하기 위하여 XMDR을 참조하여, 그 결과로 생성된 각각의 로컬 쿼리는 로컬 시스템에 의해서 수행되기 이전에 유효한 쿼리문의 형태로 바뀌게 된다. 이러한 쿼리 변환 단계는 글로벌 스키마와 로컬 스키마간의 매핑을 사용한다.

셋째, 시멘틱 변환 단계는 의미충돌을 해결한 후, 로컬 데이터 소스를 사용자가 원하는 형태로 변환시키는 과정이다. 이 과정을 통해 사용자는 자신이 원하는 뷰(view)를 통해 데이터를 보고 조작하는 것이 가능하다. 이 단계는 스키마



(그림 8) 쿼리 그래프

레벨에서 의미충돌까지도 처리할 수 있다는 장점이 있다. 네 번째, 글로벌 쿼리 결과를 조합하는 단계는 사용자에게 쿼리 결과를 보여주기 위하여 로컬 쿼리에 의해서 얻어진 쿼리 결과들을 조합한다.

4.2 쿼리 처리 예

쿼리 처리 절차를 설명하기 위해서 자산신청에 관한 비즈니스 프로세스에서 사용된 일부 쿼리 중에서 신청부분으로써 XMDR을 참조한 글로벌 스키마 기반의 사용자가 작성한 쿼리이다(그림 9).

사용자는 글로벌 스키마를 통해 입력한 (그림 9)의 점선 부분의 쿼리는 (그림 8)처럼 쿼리를 분석할 수 있다. 그러나 여기서 사용된 예제 쿼리는 (그림 9)의 점선 부분에서 두 개의 items와 applys의 관계형 테이블이 필요하기 때문에 조인연산자를 통해서 글로벌 쿼리문을 분석한 후, (그림 10)과 같이 두 개의 글로벌 스키마 그래프가 생성되며, 나중에 유효한 로컬 쿼리를 생성하기 위하여 XMDR-DQP의 의해

```
BEGIN
    IF P_APPL_EMPNO IS NOT NULL AND
          LENGTH(P_APPL_EMPNO) = 5 THEN
      SELECT 'X' INTO V_EXT
      FROM ITEMS400
      WHERE EMP_NO = P_APPL_EMPNO
      AND AUTH IN('11', '12', '13', '14');
    END IF;
END;
BEGIN
    V_DVSN_CODE := FUNC_GET_DVSN_CODE(TO_CHAR(SYSDATE,'YYYY
MMDD'),P_DEPT_CODE);

    SELECT I.NAMe, A.DEPTCODE, A.APPLYDATE, A.ITEMQUANTITY
    FROM ITEMS I, APPLY A
    WHERE A.APPLYDATE >='2009-05-13'
    AND I.NAME = '%COMPUTER5'  AND I.CODE=A.CODEID

END;
BEGIN
   INSERT INTO TAMAT110
    (DEPT_CODE, ASST_APPL_NUM, ITEM_CODE, ASST_NAME, APPL_QTY,
     APPL_EMPNO, APPL_DATE, APPL_CUSE_CODE, WRHS_RQST_DATE,
     SITE_FLAG,
     BIZ_FLAG, APPL_STTU, UPD_EMPNO, UPD_DATE, UPD_ID,
     DVSN_CODE)
VALUES
.....
EXCEPTION msg;
END;
```

(그림 9) 글로벌 스키마 기반의 비즈니스 프로세스

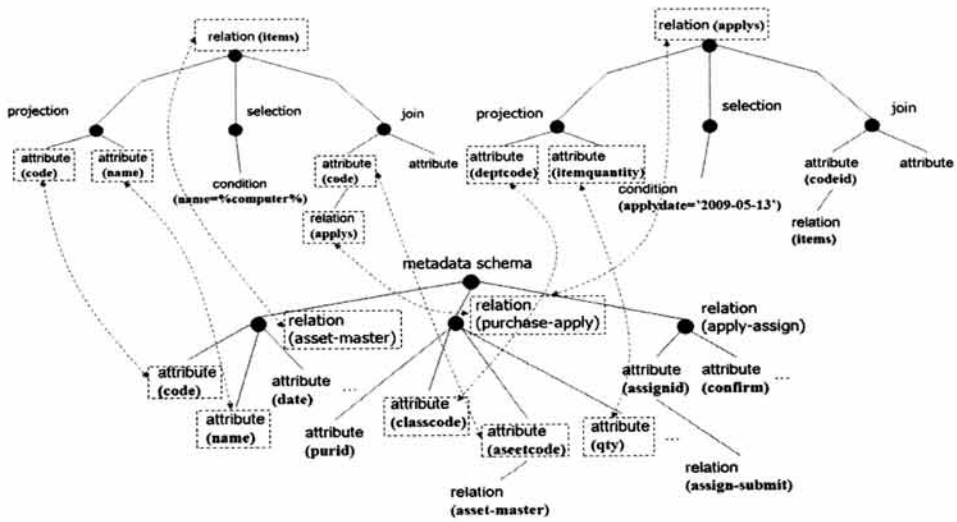
```
BEGIN
    IF P_EMPNO IS NOT NULL AND
          LENGTH(P_EMPNO) = 5 THEN
      SELECT 'X' INTO V_EXT
      FROM ASST4
      WHERE EMPNO = P_EMPNO
      AND AUTHER IN('11', '12', '13', '14');
    END IF;
END;
BEGIN
    V_DVSN_CODE := FUNC_GET_DVSN_CODE(TO_CHAR(SYSDATE,'YYYY
MMDD'),P_DEPT_CODE);

    SELECT A.NAMe, P.DEPTCODE, P.APPLYDATE, P.ITEMQUANTITY
    FROM ASSET-MASTER.A, PURCHASE. P
    WHERE P.DATE >='2009-05-13'
    AND A.NAME = '%COMPUTER5'  AND A.CODE=P.ASSETCODE

END;
BEGIN
   INSERT INTO TAMAT110
    (DEPT, APPLNUMBER, ITEM, NAME, QTY,
     EMPNO, DATE, CUSECODE, STTU, UPD,)
VALUES
.....
EXCEPTION msg;
END;
```

(그림 10) 로컬 스키마 기반의 비즈니스 프로세스

수정된다.

사용자 명세에 따라 글로벌 쿼리로 생성된 쿼리문과 그래프 예는 글로벌 쿼리를 생성한 후에 XMDR에 참조하여 실제 로컬 쿼리로 생성하는데 사용될 것이다. 그리고 (그림 10)와 같이 릴레이션의 items와 applys 두 개의 쿼리 그래프로 생성된 글로벌 쿼리는 XMDR에 등록되어 있는 로컬 스키마인 asset-master, purchase-apply, apply-assign의 릴레이션과 상호 매칭되어 로컬 쿼리로 변환되게 된다. 예를 들어 글로벌 스키마의 릴레이션인 items는 로컬 스키마에서의 asset-master로 매칭되며, projection의 속성인 code는 asset-master의 projection의 속성인 code의 매칭된다. 그리고 여기서는 두 개의 글로벌 스키마가 조인 관계이기 때문에, items에서 조인의 속성의 code의 조인 테이블이 applys이므로, 로컬 스키마의 릴레이션의 purchase-apply에 매칭된다.

이렇게 매칭이 되면 (그림 11)처럼 로컬 스키마로 변환되



(그림 11) 글로벌 쿼리와 로컬 쿼리 매핑 과정

```
<DataSourceList>
    <importedDataSource>
        <DSfactoryHandle id="s1">
                http://128.134.64.49:8080/xmdrdai/services/agent
            <SQLqueryStatement name="statement">
                <expression>
                    IF P_EMPNO IS NOT NULL AND
                      LENGTH(P_EMPNO) = 5 THEN
                      SELECT 'X' INTO V_EXT
                      FROM ASST4
                      WHERE EMPNO = P_EMPNO
                     AND AUTHER IN('11', '12', '13', '14');
                    END IF;
                </expression>
            </SQLqueryStatement>
        </DSfactoryHandle>
        <DSfactoryHandl id="s2">
                http://128.134.64.59:8080/xmdrdai/services/agent
            <SQLqueryStatement name="statement">
                <expression>
                    V_DVSN_CODE := FUNC_GET_DVSN_CODE(TO_CHAR(SYSDA
TE,'YYYYMMDD'),P_DEPT_CODE);
    SELECT A.NAMe, P.DEPTCODE, P.APPLYDATE, P.ITEMQUANTITY
    FROM ASSET-MASTER.A, PURCHASE. P
    WHERE P.DATE >='2009-05-13'
    AND A.NAME = '%COMPUTER5'  AND A.CODE=P.ASSETCODE

                </expression>
                </SQLqueryStatement>
            ..............
    </importedDataSource>
    <importedService>
        <wsdlURL>
                http://128.134.45.60:9090/axis/services/ProcessAnalyser?WSDL
        </wsdlURL>
    </importedService>
</DataSourceList>
```

(그림 12) 적용 사례에 대한 비즈니스 프로세스 문서

게 된다. 이렇게 생성된 쿼리는 협업 프로세스에 재 정의되어 사용된다. (그림 12)는 (그림 11)의 로컬 쿼리가 WSDL 문서를 기반으로 생성된 로컬 쿼리의 비즈니스 프로세스이다. 각 로컬 데이터베이스 비즈니스 프로세스 문서의 작성된 순서로 쿼리를 수행하게 된다.

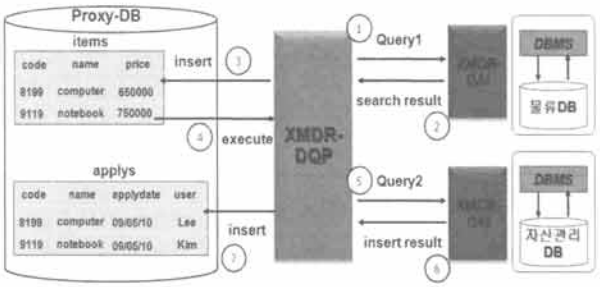
5. Proxy-DB를 이용한 데이터 이주 기법

5.1 데이터 이주

본 논문에서 제안하는 Proxy-DB를 이용한 협업에서 비즈니스 프로세스가 어떤 식으로 정의되었는지에 따라서 접근 빈도, 접근 경로, 데이터 이주의 크기가 결정된다. 일반적으로 데이터 이주는 크게 복제 및 분할로 이루어진다. 우선 복제는 원본 데이터로부터 생성되는 뷰어, 스냅샷, 가상 테이블 정의 등을 이용한 복사본이다. 그리고 분할은 데이터가 RDB(Relation DataBase)에서의 릴레이션에 스키마 변경에 의해서 수직 분할, 데이터 범주에 따른 분리 과정인 수평 분할로 나눠지며, 시스템 환경이나 데이터 서비스 목적에 알맞게 적용하는 방법에서 차이가 있다. 또한 비즈니스 프로세스에서 사용자가 작성한 쿼리 패턴에 의해서 적용되는 범위도 다르다. 따라서 협업에서 발생하는 데이터 이주는 복제, 이동, 분할 3가지로 분류할 수 있다. 이에, 본 논문에서는 Proxy-DB를 이용하여 로컬 데이터베이스 간의 협업에서 일어나는 데이터에 대한 이주를 처리한다. (그림 14)는 물류DB와 자산관리DB를 협업을 통해 자산신청을 하기 위한 프로세스인 일부분인 (그림 13)에서 Q1, Q2가 수행되는 과정을 나타낸 것이다. 우선 첫 번째 프로세스는 자산 품목 데이터를 요청하는 프로세스인 Q1이 XMDR-DQP에 의해 4장 쿼리 처리 기법에 의해 변환된 쿼리로, Q1을

```
Q1: SELECT code, name, price
    FROM items OREER BY price

Q2: INSERT INTO applys ( code, name, appludate, user )
VALUES applys ('8199', 'computer', '09/05/10', 'Lee')
INSERT INTO applys ( code, name, appludate, user )
VALUES applys ('9119', 'notebook', '09/05/10', 'Kim')
```

(그림 13) 로컬 쿼리문 예제 Q1, Q2



(그림 14) Proxy-DB를 이용한 로컬 쿼리 수행에서 발생하는 데이터 마이그레이션의 과정

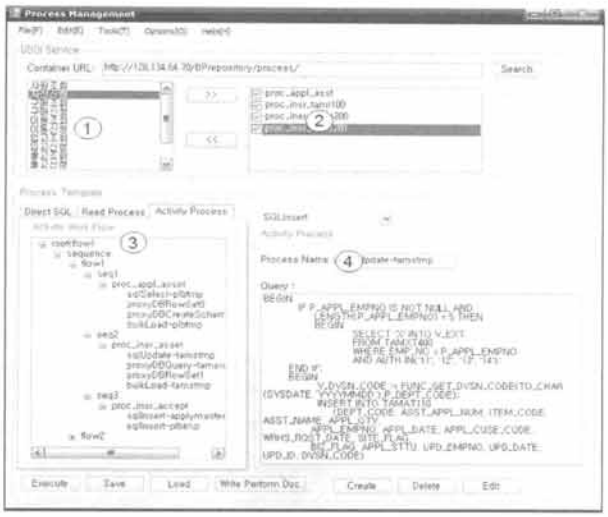
XMDR-DAI를 통해서 물류DB에서 데이터를 요청한다. 요청되어 수집된 자산 목록 데이터가 다시 XMDR-DAI와 XMDR-DQP를 통해 Proxy-DB의 items의 테이블에 삽입된다. 두 번째 프로세스인 Q2는 items의 수집된 자산 목록에서 신청할 품목을 선택한 후, XMDR-DQP와 XMDR-DAI를 통해 자산관리DB에 신청되는 쿼리이다. 신청된 후에 신청된 결과를 다시 Proxy-DB의 applys에 삽입한다. 이렇게 (그림 14)에서처럼 협업 프로세스 처리에 따라 로컬 데이터베이스 간의 데이터 이주를 처리하게 된다.

6. 적용 및 비교 분석

6.1 적용

본 논문에서의 XMDR-DSM 구현은 Windows2003, .NET 2008으로 구축하였으며, 프로세스 템플릿 레포지토리는 Oracle을 사용하여 등록하였고, Proxy-DB는 SQL 서버를 사용하였다. 적용 사례는 프로세스 관리 툴을 이용하여 통합 유형 자산 관리에서 자산 신청을 처리하는 과정에서의 일부분을 예로 하였다. (그림 15)는 사용자가 프로세스 템플릿 레포지토리의 등록된 자산 신청 프로세스를 수행하고자 하는 단위 프로시저를 선택한다.

그리고 인터페이스에서 Activity Process 탭에서 각 선택한 프로시저 수행 순서를 지정하고, 필요에 따른 프로시저 내부의 쿼리를 확인한 후, 수정을 한다. (그림 15)의 ①은 웹 서비스(UDDI)에서 제공하는 비즈니스 프로세스 목록이고, 자산신청을 선택하면, ②에서 선택한 프로세스의 내부 프로시저를 선택할 수 있다. 이 내부 프로시저는 자산신청이라는 비즈니스 프로세스 수행되기 위한 세부적인 단위 프로세스이다. ③은 선택한 자산신청 프로세스의 내부 프로시



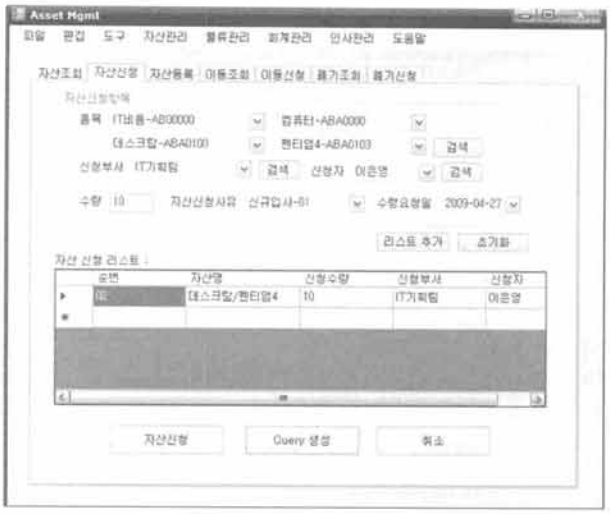
(그림 15) 인터페이스 1

(그림 16) 인터페이스 2

저의 수행 순서를 정의한 것이고, ④는 프로시저 하나하나의 글로벌 스키마 기반의 쿼리문이다. (그림 15)의 과정을 거치면, (그림 16)과 같이 자산신청 프로세스가 실행되어 결과 데이터를 저장하여 데이터 처리 및 가공할 Proxy-DB를 접근하여 테이블을 생성하는 인터페이스이다. Proxy-DB에 새롭게 생성할 테이블명, 필드명, 타입, 크기 등을 정한다. 그리고 화면에서처럼 Proxy-DB Fields에서 해당 테이블 스키마 항목을 확인할 수 있다.

(그림 17)은 적용사례인 통합 유형 자산 관리 시스템에서 자산 신청을 하는 화면이다. 자산신청에 필요한 항목을 선택 및 입력한 후에 리스트 추가를 하고, Query를 생성하면 글로벌 스키마 기반의 자산 신청 쿼리문이 생성되고, 이를 수행할 수 있게 구성되어있다.

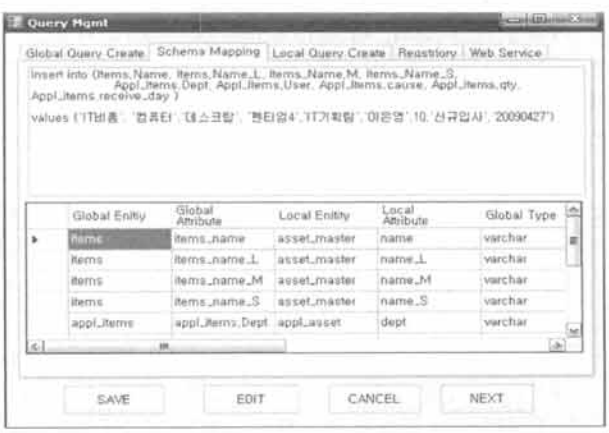
(그림 17)에서 생성된 자산 신청 글로벌 쿼리는 (그림 18)에서처럼 스키마 내용을 확인하고, 변경이 가능하다. 이렇게 생성된 쿼리는 (그림 19)에서처럼 로컬 스키마와 매핑하게



(그림 17) 인터페이스 3

(그림 18) 인터페이스 4

되고, 매핑에 대한 쿼리를 확인한 후에는 최종 로컬 데이터베이스에 실행될 로컬 쿼리를 (그림 20)처럼 생성된다. 이와



(그림 19) 인터페이스 5

(그림 20) 인터페이스 6

같은 절차로 작성된 로컬 쿼리는 각 로컬에 수행되게 되고, 새롭게 추가되거나 변경된 사항에 대해서는 프로세스 템플릿 레포지토리에 등록되어진다. (그림 20)처럼 최종 작성된 쿼리들은(단위 프로시저 기반의 프로세스) XML 기반의 WSDL로 생성되어 수행된다.

6.2 타 시스템 비교 분석

본 논문에서는 전사적 데이터 통합 서비스를 제공하기 위해서 협업을 위한 시스템 구축을 하는데 있어서 필요한 미들웨어인 XMDR-DSM을 제안하였다. 이에 따라 OGSA-WebDB[13], Grid-DBMS[12]와 비교 분석한 내용은 (표 1)과 같다. 통합 유형 자산 관리 시스템에 적용한 XMDR-DSM 시스템은 기존의 자산 정보를 관리하는 로컬 시스템들의 협업을 지원하도록 구축하였다. 제안한 XMDR-DSM 시스템을 적용하여 시스템 상호운용성과 데이터 상호운용성 측면에서 Grid-DBMS와 OGSA-WebDB를 적용할 경우에 비하여 부분적으로 향상시킬 수 있으며, 이를 토대로 로컬 시스템들이 효율적으로 데이터를 상호 운용할 수 있다. 우선 시스템 상호운용성은 기존의 로컬 시스템들이 운영하는 데이터를 유실되지 않도록 하고, 데이터의 연속성을 유지시켜야 하며, 기존 비즈니스 프로세스 지원에 대한 확장이 유연해야 한다. 이는 협업에서 필요한 데이터를 분석하여 상호간의 업무처리가 가능해야 되기 때문이다. 따라서 통합 유형 자산 관리 시스템에서 최근에 추가된 자산의 정보를 회계 관리 시스템이나 통계 시스템 등에 비즈니스 프로세스를 기반의 협업에 활용할 수 있도록 되어야 한다. Grid-DBMS는 데이터 대한 동적관리는 지원을 하지만 비즈니스 프로세스 수준의 기술 지원은 고려되지 않았다. 또한 데이터의 교환 기술로 Grid-FTP를 이용하여 데이터를 복사할 수 있도록 하였지만, 이를 지원하지 않는 로컬 시스템에서는 데이터 교환이 용이하지 않다. 또한 통합 유형 자산 관리 시스템에 입고 관리 시스템을 추가하여 확장을 진행할 경우에 자산 정보에 대한 의미와 표현 형태가 동일하게 표준화되어야 한다. 협업 환경에서 표준화는 시스템마다 다르게 표현된 데이터나 메타데이터에 대하여 동일하게 활용할 수 있도록 하는 것이다.

따라서 추가되는 시스템과의 데이터 이질성을 극복할 수 있도록 표준화 작업이 필요하다. 예를 들면, 로컬 시스템인 입고 관리 시스템이 협업에 참여하고자 하면 통합 유형 자산 관리 시스템에서 사용하는 데이터 교환 및 공유 표준에 따라야 한다. 이에 대하여 중앙 집중 관리하는 OGSA-WebDB는 데이터 접근방식을 중개방식인 어댑터 유형으로

〈표 1〉 타 시스템과 비교 분석

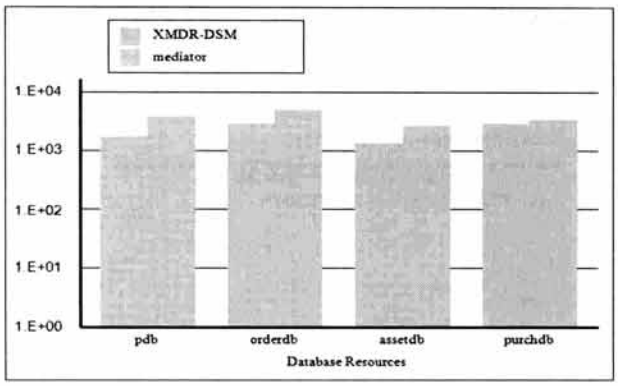
×:낮음 △:보통 ○:높음

			Grid-DBMS	OGSA-WebDB	본 시스템
시스템 상호운용성	비즈니스 프로세스 지원여부	설명	데이터에 대한 동적관리는 지원하지만, 비즈니스 프로세스 지원하지 않음.	데이터를 분석하여 의사결정 프로세스는 지원하지만, 질의기반의 비즈니스 레벨의 지원은 하지 않음.	데이터의 교환 및 공유를 목적으로 한 워크플로우 기반의 비즈니스 프로세스 지원함.
		비교	×	△	○
	교환기술	설명	Grid-FTP를 지원하는 로컬 시스템에서만 데이터 교환 가능함.	로컬 시스템들 간의 데이터 접근은 지원하지 않고, Globus 미들웨어 기반의 WebDB를 이용한 접근만 가능함.	로컬 시스템 어느 곳에서나 XMDR-DSM 이용한 데이터 접근 및 교환이 가능함
		비교	△	○	○
데이터 상호운용성	이질성 해결	설명	데이터복사 기능을 이용하여 구문, 구조 이질성 해결	WebDB 시스템의 메타데이터 스키마를 이용하여 구문, 구조 이질성 해결	XMDR을 이용하여 데이터의 구문, 구조, 의미적 이질성 극복
		비교	△	△	○
	일관성 유지	설명	데이터 복사, 단편화 기능을 이용하여 일관성을 유지하지만 시스템간의 변경 지연 발생	중앙 집중 관리 저장소인 GDS를 이용하여 실시간으로 데이터 변경 반영하여 일관성 유지(From절 없는 질의문 이용)	로컬 시스템에서 발생하는 데이터 변경은 이주 서비스를 이용하여 일관성 유지(글로벌 스키마를 이용)
		비교	△	○	○

구축되었기 때문에 시스템 확장에 따른 유연함이 부족하다. 협업에 참여하기 위한 로컬 시스템이 증가할수록 스키마와 다른 로컬 시스템과의 연계를 새로이 구축해야하는 단점이 있다. 본 시스템은 기존의 로컬 시스템의 변경을 최소화하기 위하여 에이전트 유형으로 데이터를 접근하며 데이터 교환 및 공유를 위한 표준화 작업을 통해 시스템 확장을 지원하고 데이터 일관성 처리를 위하여 분산된 환경을 고려하였다. Grid-DBMS는 이종 데이터를 접근하기 위하여 가상공간을 제공하여 부분적으로 구문적, 구조적 이질성을 극복할 수 있다.

하지만 데이터를 전송하는 기술인 Grid-FTP로 인하여 데이터들의 의미적인 연관성은 고려되지 않았으며 Grid-FTP를 지원하는 로컬 시스템에서만 활용이 가능하다. OGSA-WebDB는 허브 시스템에서 지원하는 메타데이터를 이용하여 데이터의 구문적, 구조적 이질성을 극복할 수 있으며 모든 로컬 시스템에서 사용이 가능하도록 XML 메시지를 사용한다. 그러나 OGSA-WebDB는 중앙 집중 관리 방식이기 때문에 로컬 시스템 간의 데이터를 활용하기 위해서는 GDS를 이용해야 한다. GDS는 로컬 시스템들의 데이터를 중개하는 역할을 수행하기 때문에 부분적으로 데이터의 의미적 이질성은 해결할 수 없으며 데이터를 정적으로 활용할 수밖에 없다. 본 시스템은 데이터 교환 및 공유의 표준인 XMDR을 이용하여 데이터의 구문적, 구조적, 의미적 이질성을 해결하고 글로벌 스키마를 기반으로 데이터를 교환하므로 모든 로컬 시스템에서 데이터를 활용할 수 있도록 하였다. 특히 본 시스템은 XMDR-DSM을 이용하므로 많은 기능을 확장시킬 수 있다. 그러나 본 시스템은 로컬 시스템들의 데이터를 분석하는 모듈을 지원하지 않는다. 이는 데이터가 독립적으로 분산 운영되는 환경을 고려하였기 때문에 데이터의 중앙 집중처리를 지원할 수 없다. 또한 대용량의 데이터 처리에 따른 데이터 중복 문제를 내포하고 있기 때문에 본 시스템은 로컬 시스템들의 협업에 한하여 활용될 수 있다. 따라서 실시간 기업환경에서 요구되는 데이터 분석을 지원할 수 있는 모듈과 대용량 데이터 처리를 지원하기 위하여 향후 계속적인 연구가 필요하다.

본 시스템에서 XMDR-DSM를 이용한 통합 유형 자산관리 시스템에서 비즈니스 프로세스 내부의 쿼리문 처리에 따른 데이터 접근, 수집, 통합 서비스를 수행할 때 처리 수행 시간에 대해서 (그림 21)과 같이 성능을 평가한다. 성능 평가를 위한 테스트는 협업된 4개의 데이터베이스 서버를 이용하였고, 약25,000건 이상의 결과를 가지는 SQL Query 문을 사용하였다. (그림 21)는 XMDR-DSM을 이용한 경우와 일반 Grid-DBMS나 OGSA-WebDB 기술을 이용한 mediator를 이용하여 Query의 처리 수행 시간을 비교하였다. pdb는 mediator를 이용한 Query 처리 시간이 약2150ms 정도의 시간이 걸린데 반에, XMDR-DSM를 이용한 Query 처리 시간이 약6520ms 정도 걸린다. orderdb도 약5210ms와 약7500ms 시간이 소요된다. assetdb도 mediator처리 시간이 약1200ms에 비해 XMDR-DSM의 처리 시간이 약3900ms,



(그림 21) 쿼리 수행 처리 시간 비교

purchdb역시 약4500ms와 약5100ms 시간이 소요된다. 따라서 XMDR-DSM를 이용한 처리 시간이 mediator를 이용한 처리 시간보다 약600~4000ms 정도 더 소요되지만, 이는 각 데이터베이스 서버마다 네트워크에서 소요되는 시간, 각 데이터베이스 서버의 에이전트에서 소요하는 시간, XMDR-DB와 Proxy-DB에 접근하는데 소요되는 시간이다. (그림 21)에서처럼 테스트한 결과에 따르면 각 소요되는 시간이 거의 일정하고, 각 요소마다 처리되는 시간이 전체 시간에 비해 미약하다. 하지만 XMDR-DSM은 비즈니스 프로세스가 시스템에 의해 자동으로 수행되고, 이에 따른 프로세스를 업무에 따라 중복을 방지하여 반복적으로 모니터링하고 관리해 나감으로써 협업에 필요한 비즈니스 프로세스를 개선해 나갈 수 있다. 또한 전사적 비즈니스 프로세스의 개선을 위한 정확한 정보를 제공해 줌으로써 프로세스의 운영을 더욱 개선할 있는 기회를 제공할 수 있는 이점이 있다.

7. 결 론

본 논문에서는 비즈니스 프로세스를 통해 상호간의 업무 지원이 가능한 협업을 위한 XMDR-DSM 시스템을 제안하였다. 또한 비즈니스 프로세스의 처리 과정이 기술된 워크플로우를 이용한 협업을 효율적으로 지원할 수 있게 로컬 시스템간의 데이터 상호운용상의 브리지 역할을 하고, 협업 프로세스에 따른 Data Federation, Data Migration, Data Synchronization, Data Quality, Data Integration 이 가능하게 고려하였다.

따라서 본 논문에서 제시한 데이터베이스간 협업을 통해 비즈니스 프로세스를 처리할 수 있게 환경을 구축하였고, 협업 프로세스 처리시 발생하는 이질적인 데이터간의 상호운용이 가능하게 하는 효율적인 서비스를 지원하였다. 그리고 비즈니스 프로세스 재 정의에 의한 처리, 새롭게 정의되는 프로세스를 사용자가 작성하여 처리할 수 있게 하였다. 이는 분산 데이터의 연속성, 접근성을 바탕으로 데이터 이질성을 해결하여 일관성을 유지하면서, 공유 및 교환되는 데이터의 활용을 지원하여 가용성, 투명성을 향상시킬 수

있는 협업 프레임워크를 구축하였다. 이는 데이터 수집을 지원하고, 데이터의 의미적 연관성을 고려한 검색이 보다 효율적으로 활용할 수 있었다. 또한 제안한 XMDR-DSM을 이용한 협업 시스템은 비즈니스 프로세스를 통해 이 기종 데이터베이스 상호간의 업무 지원이 가능해지고, 정보를 재활용 가능한 자산으로 확보할 수 있는 이점과 정보 제공 및 교환의 표준화를 제시할 수 있다. 그리고 워크플로우 기반의 비즈니스 프로세스를 통해 일원화된 정보를 관리할 수 있게 된다. 따라서 정보 수명을 증대하고 비용절감을 할 수 있게 되었다. 이후에는 기업 정보시스템인 ERP, EAI 등에서 확장되어 워크플로우 프로세스 백본의 역할로 수행되는 통합 프로세로 확장하는 연구가 필요하다.

참 고 문 헌

[1] E. Al-Masri and Q.H. Mahmoud, "Crawling Multiple UDDI Business Registries," Proc. 16th Int'l World Wide Web Conf., ACM Press, pp.1255-1256, May, 2007.

[2] Fred A. Cummins, "Enterprise Integration: An Architecture for Enterprise Application and Systems Integration", Wiley; 1st edition, pp.496, February, 1, 2002.

[3] E. Al-Masri and Q.H. Mahmoud, "A Framework for Efficient Discovery of Web Services across Heterogeneous Registries," Proc. IEEE Consumer Communications and Networking Conf. (CCNC), IEEE CS Press, pp.415-419, Jan., 2007.

[4] Eyhab Al-Masri, Qusay H. Mahmoud, "Interoperability among Service Registry Standards", Published by the IEEE Computer Society, pp.210-220, June, 2007.

[5] UDDI, UDDI Technical White Paper, http://www.uddi.org/

[6] Arpinar I.B., Aleman-Meza B., Zhang R., and Maduko A., "Ontology-Driven Web Services Composition Platform", IEEE Conference on E-Commerce Technology(CEC 2004), San Diego, California, pp.175-199, July, 2004.

[7] Mehmet Sayal, Fabio Casati, Umesh Dayal, Ming-Chien Shan., "Integrating Workflow Management Systems with Business-to-Busines-s Interaction Standards", Proc. 18th International Conference on Data Engineering (ICDE'02), pp.287 2002.

[8] Kevin D. Keck and John L. McCarthy, "XMDR: Proposed Prototype Architecture Version 1.01", http://www.xmdr,irg, February, 2005.

[9] Youn-Gyou Kook, Gye-Dong Jung, Young-Keun Choi, "Data Grid System Based on Agent for Interoperability of Distributed Data", LNAI, 4088, pp.162-174, Aug., 2006.

[10] 문석재, 정계동, 강석중, 최영근, "저장-프로시저 기반의 비즈니스 프로세스 상호운용을 위한 XMDR Hub 프레임워크", 한국해양정보통신학회, 제12권 12호, pp.2207-2218, 2008.

[11] 문석재, 정계동, 최영근, "분산 데이터 상호운용을 위한 SQL/XMDR 메시지 기반의 Wrapper를 이용한 데이터 허브 시스템", 한국해양정보통신학회, 제11권 11호, pp.2047-2058, 2007.

[12] G Aloisio, M Cafaro, S Fiore, M Mirto, "The Grid-DBMS: Towards Dynamic Data Management in Grid Environments", Information Technology:Coding and Computing(ITCC'05), IEEE, Vol.2, pp.199-204, April, 2005.

[13] Said Mirza Pahlevi, Isao Kojima, "OGSA-WebDB: An OGSA-Based System for Bringing Web Databases into the Grid," itcc, Vol.2, pp.105-109, 2004.

[14] Globus Toolkit, .http://www-unix.globus.org/toolkit/.12

[15] GridFTP Protocol, "http://www-fp.mcs.anl.gov/ds1/GridFTP-Protocol-RFC -DRAFT.pdf"

[16] ISO/IEC IS 11179, "Information technology -Specification and standardization of data elements", 2003.

[17] Thatte S., XLang: Web Services for Business Process Design, http://www.gotdotnetcom/team/xml_wsspecs/xlang-c/default.htm, 2002.

[18] OGSA-DAI, http://www.ogsadai.org.uk/

[19] Open Grid Services Architecture Data Access and Integration (OGSA-DAI). http://www.ogsa-dai.org/

[20] Antonioletti, M., Atkinson, M, Baxter, R., Borley, A., Chue Hong, N., Collins, B., Hardman, N., Hume, A., Knox, A., Jackson, M., Krause, A., Laws, S., Magowan, J., Paton, N., Pearson, D., Sugden, T., Watson, P. and Westhead, M., "The design and implementation of Grid database services in OGSA-DAI". Concurrency and Computation: Practice and Experience, 17(2):, pp.357-376, 2005.

문 석 재

e-mail : msj8086@gmail.com

2002년 독학사 전자계산학과(이학사)

2004년 광운대학교 컴퓨터소프트웨어학과 (석사)

2004년~2005년 필컴정보시스템 주임연구원

2006년~현 재 광운대학교 컴퓨터과학과 박사과정

관심분야 : XMDR, 데이터 그리드, 상호운용성

정 계 동

e-mail : gdchung@kw.ac.kr
1985년 광운대학교 전자계산학
1992년 광운대학교 산업정보학(석사)
2000년 광운대학교 컴퓨터과학(박사)
1993년~2004년 광운대학교 정보과학원 교수
2005년~현 재 광운대학교 교양학부 교수
관심분야 : XML 분산시스템, 분산 컴퓨팅기술, 이동에이전트

최 영 근

e-mail : ygchoi@kw.ac.kr
1980년 서울대학교 수학교육과(이학사)
1982년 서울대학교 계산통계학과(이학석사)
1989년 서울대학교 계산통계학과(이학박사)
1983년~현 재 광운대학교 컴퓨터과학과 교수
1992년~2000년 광운대학교 전산정보원 원장
2002년~2005년 광운대학교 교무연구처장
관심분야 : 객체지향설계, 분산시스템, 이동에이전트, 상호운용성