

# 메타데이터 레지스트리 기반의 분산 정보 통합 시스템 설계 및 구현

김 종 환<sup>†</sup> · 박 혜 숙<sup>††</sup> · 문 창 주<sup>†</sup> · 백 두 권<sup>†††</sup>

## 요 약

중개기 기반 정보 통합 시스템은 서로 다른 지역 정보 시스템의 유연한 통합을 지원하나, 질의 처리시 최적화 측면과 지역 스키마 정보에 관한 메타데이터 표준화 측면에는 그리 큰 비중을 두지 않았다. 이러한 점을 개선하기 위해 제안된 분산 정보 통합 시스템은 질의 처리시 최적화 측면을 위해 질의 캐싱을 사용하며, 지역 스키마 정보에 관한 메타데이터 표준화 측면을 위해 ISO/IEC 11179 기반의 메타데이터 레지스트리를 사용한다. 이 시스템은 분산된 이기종의 비즈니스 정보 시스템들을 논리적으로 통합하여 사용자가 필요로 하는 통합된 정보를 웹 기반으로 제공한다. 이러한 시스템을 시스템 재사용성의 향상과 유지보수의 용이함을 위해 계층적 패턴을 사용하여 3계층 표현 방식 아키텍처로 표현하였고, 3계층 아키텍처의 핵심 요소들의 기능성과 흐름을 효과적으로 표현하기 위하여 UML 방법론을 확장한 EPEM 방법론을 이용하여 설계하였다. 또한 제안된 시스템의 구체적인 한 예로서, 공급망 관리 도메인에 적용하여 웹 기반으로 구현하였다. 따라서 분산 정보 통합 시스템은 질의 처리 속도 향상을 위해 질의 함수 관리기와 질의 함수 저장소를 통하여 질의 캐싱 기능을 제공하였고, 의미 이질성 해결을 위해 ISO/IEC 11179 기반의 메타데이터 레지스트리와 스키마 레파지토리를 이용함으로써 스키마 이질성과 데이터 이질성을 해결하였다.

## Design and Implementation of A Distributed Information Integration System based on Metadata Registry

Jong-Hwan Kim<sup>†</sup> · Hea-Sook Park<sup>††</sup> · Chang-Joo Moon<sup>†</sup> · Doo-Kwon Baik<sup>†††</sup>

## ABSTRACT

The mediator-based system integrates heterogeneous information systems with the flexible manner. But it does not give much attention on the query optimization issues, especially for the query reusing. The other thing is that it does not use standardized metadata for schema matching. To improve this two issues, we propose mediator-based Distributed Information Integration System (DIIS) which uses query caching regarding performance and uses ISO/IEC 11179 metadata registry in terms of standardization. The DIIS is designed to provide decision-making support, which logically integrates the distributed heterogeneous business information systems based on the Web environment. We designed the system in the aspect of three-layer expression formula architecture using the layered pattern to improve the system reusability and to facilitate the system maintenance. The functionality and flow of core components of three-layer architecture are expressed in terms of process line diagrams and assembly line diagrams of Eriksson Penker Extension Model (EPEM), a methodology of an extension of UML. For the implementation, Supply Chain Management (SCM) domain is used. And we used the Web-based environment for user interface. The DIIS supports functions of query caching and query reusability through Query Function Manager (QFM) and Query Function Repository (QFR) such that it enhances the query processing speed and query reusability by caching the frequently used queries and optimizing the query cost. The DIIS solves the diverse heterogeneity problems by mapping MetaData Registry (MDR) based on ISO/IEC 11179 and Schema Repository (SCR).

**키워드 :** 데이터베이스(Database), 분산 정보 통합 시스템(Distributed Information Integration System), 메타데이터 레지스트리(Metadata Registry), 스키마 레파지토리(Schema Repository), 이질성(Heterogeneity), 질의 함수 저장소(Query Function Repository), EPEM, 공급망 관리(SCM), 질의 캐싱(Query Caching), 질의 재사용성(Query Reusability)

### 1. 서 론

기업 집합 환경 하에서의 대부분의 정보 시스템들이 속성

상 지역적으로 분산되어 있으며, 다양한 이기종의 데이터 소스들로 구성되어 있기 때문에, 지역적으로 분산된 이기종의 데이터 소스들로부터 사용자 의사 결정을 지원하는데 필요한 통합된 정보를 얻는 것은 어려운 것이 사실이다. 따라서 분산된 시스템들이 서로 간의 협조 하에 자신들의 역할을 수행하도록 하고, 분산된 컴퓨터들 간의 데이터 교환과 다양한 종류의 데이터베이스에 대한 접근이 가능하도록 이들을

\* 본 연구는 2001년 고려대학교 특별연구비와 2002년 한국과학재단 목적기초 연구 지원으로 이루어졌습니다.

† 준 회원 : 고려대학교 대학원 컴퓨터학과

†† 정 회원 : 고려대학교 대학원 컴퓨터학과

††† 종신회원 : 고려대학교 컴퓨터학과 교수

논문접수 : 2002년 7월 24일, 심사완료 : 2002년 12월 12일

체계적이고 효율적으로 통합하는 것은 매우 중요하다[1].

이러한 문제를 효과적으로 해결하기 위해서는 지역적으로 분산된 데이터베이스에 대한 접근뿐만 아니라 다양한 사용자들이 서로 이질적인 데이터베이스를 투명성을 가지고 편리하게 사용할 수 있도록 하는 분산 통합 기술이 필요하다. 분산 통합 기술은 다양한 형태의 데이터베이스와 자료들을 효율적으로 통합하여 사용자에게 단일의 인터페이스를 제공하는 분산 통합 데이터베이스를 가능하게 한다. 분산 통합 데이터베이스란 기존의 시스템과 데이터가 통합된 형태의 데이터베이스로, 여기서의 통합이란 다양한 형태의 정보를 어디에서나 투명하게 실시간에 접근할 수 있다는 의미의 논리적 통합을 말한다[2].

따라서 본 논문에서 제안한 분산 정보 통합 시스템은 기업 집합 환경 하에서 운영되고 있는 분산된 이기종의 비즈니스 정보 시스템들을 논리적으로 통합하여, 사용자 의사 결정을 지원하는데 필요한 통합된 정보를 웹 상에서 제공한다. 이 시스템은 질의 함수 저장소, 메타데이터 레지스트리, 스키마 레퍼지토리를 이용하여, 각기 기 구축된 데이터베이스의 독립성을 최대한 유지하면서, 이질적이고 분산되어 있는 데이터베이스를 마치 하나의 데이터베이스 처럼 단일의 인터페이스를 가지고 투명성 있게 접근하는 것을 가능하게 한다.

이러한 분산 정보 통합 시스템이 복잡한 비즈니스 프로세스와 다양한 리소스들을 포함하고 있기 때문에, 이러한 특성들을 효율적으로 표현하기 위하여 시스템의 핵심 기능을 담당하는 관리기들에 대하여 UML 방법론[3]을 확장한 EPEM(Eriksson Penker Extension Model) 방법론[4]을 이용하여 그것들의 기능성과 흐름을 설계하였다. EPEM 방법론이란 복잡한 비즈니스 프로세스와 다양한 종류의 리소스를 좀 더 명확하게 식별하고 가시화하기 위하여 UML 방법론에 프로세스 라인 다이어그램(Process Line Diagram)과 어셈블리 라인 다이어그램(Assembly Line Diagram)을 추가한 방법론이다. 또한 제안한 시스템의 한 구체적인 예로서, 공급업체, 제조업체, 유통업체, 소매업체, 소비자가 연계된 공급망 관리(SCM: Supply Chain Management) 도메인의 '수주 관리' 부분에 적용하여 웹 상에서 동작할 수 있도록 구현하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 분산 정보 통합 시스템과 관련된 이슈에 대해서 설명한다. 3장에서는 메타데이터 레지스트리를 기반으로 하는 분산 정보 통합 시스템의 3계층 아키텍처를 제시하고, 제시된 아키텍처의 각각의 기능성과 흐름을 표현하기 위하여 EPEM 방법론을 이용한 설계를 다룬다. 4장에서는 제안된 분산 정보 통합 시스템의 구체적인 한 예로서, 웹 기반의 공통된 사용자 인터페이스를 가지고 사용자가 통합된 질의를 할 수 있도록 공급망 관리 도메인에 적용하여 구현하였다. 5장에서는 메타데이터 레지스트리 기반의 분산 정보 통합 시스템을 다른 정보 통합 시스템들과 비교 평가한다. 마지막으로 6장에서

는 결론 및 향후연구에 대해서 설명한다. 부록에서는 EPEM 방법론의 프로세스 다이어그램, 어셈블리 라인 다이어그램, 표기법에 대해서 설명하였다.

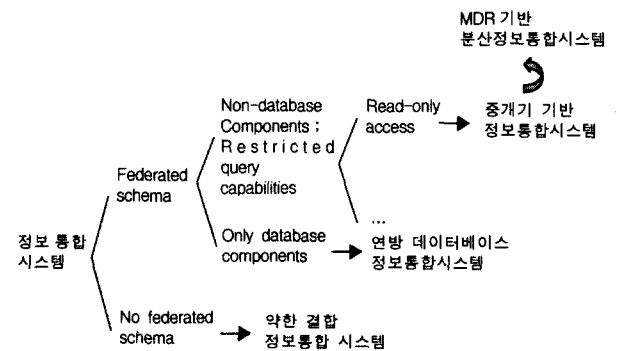
## 2. 관련 연구

본 장에서는 분산 정보 통합 시스템과 관련된 이슈들을 설명한다. 먼저, 정보 통합 시스템의 분류에 대해서 살펴보고, 데이터 통합시 발생하는 이질성에 대해서 살펴본 뒤, 데이터의 원활한 유통과 공유를 가능하게 하는 메타데이터 레지스트리에 대해서 살펴보도록 한다.

### 2.1 정보 통합 시스템 분류

데이터베이스 통합은 분산 데이터베이스 환경에서 각 지역 데이터베이스를 전역의 가상 스키마로 통합하는 데이터베이스 스키마 통합과 서로 다른 사용자 뷰를 통합하는 뷰 통합의 두 영역으로 구분된다[2]. 먼저, 데이터베이스 스키마 통합은 통합 데이터베이스 시스템에서 사용자에게 투명성을 제공하기 위해 중요하며, 스키마 통합 정도에 따라 강한 결합(tightly-coupled) 방식과 약한 결합(loosely-coupled) 방식으로 구분된다[5]. 강한 결합 방식은 구성 데이터베이스의 스키마를 전역 스키마로 통합하는 방식으로 METU의 MIND [6] 등에서 활용되었으며, 약한 결합 방식은 스키마 통합을 배제하는 방식으로 Litwin[7] 등에서 활용되었다. 다음으로, 데이터베이스 뷰 통합은 여러 응용의 모든 요구사항을 전체에 대한 개념적 스키마로 합병하는 것이다. 뷰는 응용에서의 요구사항을 나타내는 개념적 스키마를 말한다. 이러한 뷰 통합시에는 이질성이 주된 문제로 나타난다.

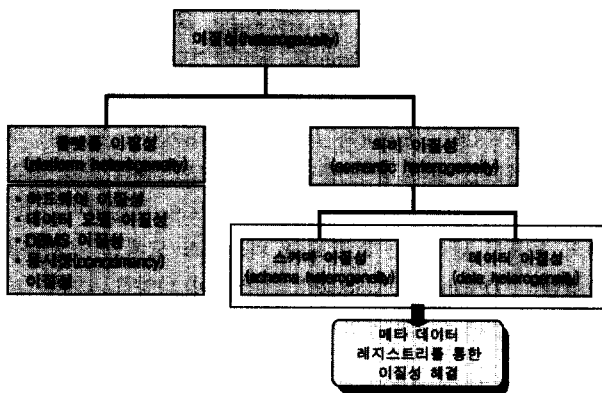
이러한 데이터베이스 통합에 대한 정보 통합 시스템은 (그림 1)과 같이 분류된다. 중개기 기반 정보 통합 시스템은 서로 다른 지역 정보 시스템의 유연한 통합을 지원하나, 질의 처리시 최적화 측면과 지역 스키마 정보에 관한 메타데이터 표준화 측면에는 그리 큰 비중을 두지 않았다[8]. 본 논문에서 제안한 분산 정보 통합 시스템은 중개기 기반 정보 통합 시스템의 범주에 속한다.



(그림 1) 정보 통합 시스템의 분류

2.2 데이터 통합시의 이질성 분류

둘 이상의 데이터베이스 시스템이 효과적으로 상호 운영 되도록 하는데는 많은 해결해야 할 문제들이 존재한다. 가장 기본적이고 주된 문제는 이질성(heterogeneity) 문제이다. (그림 2)는 이질성의 분류와 본 논문에서 해결하고자 하는 이질성 범주에 대해서 보여준다. 플랫폼 이질성(platform heterogeneity)은 하드웨어 이질성, 데이터 모델 이질성, DBMS 이질성, 동시성 이질성 등을 포함한다. 반면 의미 이질성(semantic heterogeneity)은 데이터의 논리적 표현에 있어서의 이질성이며, 개별 데이터베이스들이 서로 다른 스키마를 가지고 서로 다른 데이터 정의를 가지기 때문에 발생한다. 스키마 이질성(schema heterogeneity)은 주로 동일한 정보에 대해 다른 구조를 사용하고, 동일한 구조에 대해서 서로 다른 개체명이나 컬럼명을 사용함으로써 야기되며, 데이터 이질성(data heterogeneity)은 데이터 포맷의 불일치로 야기된다[9].



(그림 2) 이질성 분류와 본 논문에서 해결하고자 하는 이질성 범주

2.3 메타데이터 레지스트리(MDR : MetaData Registry)

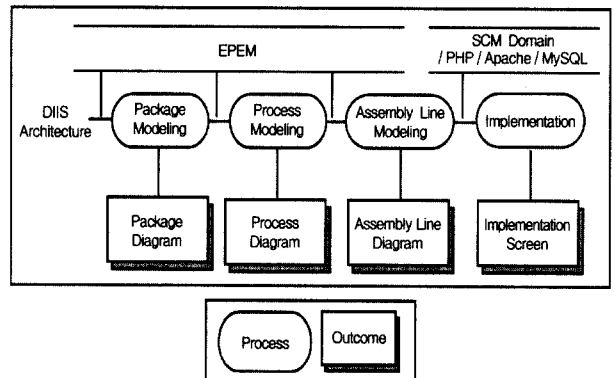
메타데이터 레지스트리에 대해서는 ISO 11179에서 정의하고 있다[10]. ISO 11179는 메타데이터 즉, 데이터 요소의 명세와 표준화를 위하여 데이터의 요청과 등록을 유용화 하고, 데이터의 접근과 사용을 촉진하기 위하여 데이터를 이해할 수 있고, 공유할 수 있도록 만들기 위한 표준과 등록에 대한 내용을 설명하고 있다.

메타데이터 레지스트리는 메타데이터의 등록과 인증을 통하여 표준화된 메타데이터를 유지 관리하며, 메타데이터의 명세와 의미의 공유를 목적으로 한다. 이것은 데이터 공유를 위한 기본 틀로서 공유를 위한 기본 요소로 데이터 요소를 사용하고, 이러한 데이터 요소가 그 내포적인 의미를 충분히 나타내어 줄 수 있는 구조내에 등록하여 각 응용에 활용토록 제공한다. 상이한 시스템간 또는 조직간의 데이터 공유를 제공하기 위하여 어떤 특정 분야의 데이터 요소에 대한 특성 정보로 추출된 메타데이터들을 등록하여 모아

놓은 참조 정보의 집합체이다. 이러한 MDR은 관리 부분(stewardship region), 명명과 식별 부분(naming & identification region), 분류 부분(classification region), 데이터 요소 관리 부분(data element administration), 데이터 요소 개념 관리 부분(data element concept administration), 개념 또는 값 영역 관리 부분(concept or value domain administration)의 여섯 부분으로 구성된다[11].

3. 분산 정보 통합 시스템의 설계

제안된 분산 정보 통합 시스템(DIIS : Distributed Information Integration System)은 기업 집합 환경 하에서 운영되고 있는 분산된 이기종의 비즈니스 정보 시스템들을 물리적 통합이 아닌 논리적으로 통합함으로써 사용자 의사 결정을 지원하는데 필요한 통합된 정보를 웹 기반으로 제공하는데 목표를 두고 있다. 이러한 시스템의 설계 및 구현 과정은 (그림 3)과 같다.



(그림 3) MDR 기반의 분산 정보 통합 시스템 설계 및 구현 과정

본 장에서는 메타데이터 레지스트리를 기반으로 하는 분산 정보 통합 시스템의 3계층 아키텍처를 제시하고, 제시된 아키텍처의 각각의 기능성과 흐름을 표현하기 위하여 UML 방법론을 확장한 EPEM 방법론을 이용하여 설계하는 것을 다룬다. 다음 장에서는 제안된 시스템의 구체적인 한 예로서, 공급망 관리 도메인의 '수주 관리' 부분에 적용하여 구현함으로써 제안된 시스템이 이질성을 해결하는 것과 질의 개선되는 것을 보인다.

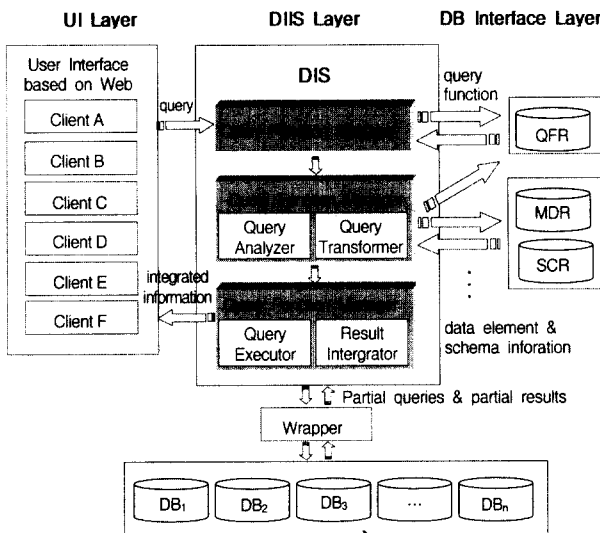
3.1 분산 정보 통합 시스템 구성

각기 기 구축된 데이터베이스의 독립성을 최대한 유지하면서 이질적이고 분산되어 있는 데이터베이스를 마치 하나의 데이터베이스 처럼 투명성 있게 사용할 수 있는 분산 정보 통합 시스템은 크게 네 부분으로 구성된다.

- ① 웹 기반의 사용자 인터페이스(User Interface based on Web) : 웹 기반 사용자 질의의 입출력을 담당

- ② 분산 정보 통합 시스템(DIIS : Distributed Information Integration System) : 입력된 사용자 질의를 질의 함수를 이용하여 분석, 변환, 수행, 통합함으로써 사용자가 원하는 통합된 정보를 제공
- ③ 질의 함수 저장소(QFR : Query Function Repository), 메타데이터 레지스트리(MDR : MetaData Registry), 스키마 레파지토리(SCR : Schema Repository) : 질의 함수, 데이터 요소, 스키마 정보를 저장
- ④ 분산된 데이터베이스 : 공급업체, 제조업체, 유통업체, 소매업체, 소비자 등의 실제 데이터를 저장

3.2 MDR 기반의 분산 정보 통합 시스템의 3계층 아키텍처  
 분산 정보 통합 시스템의 구성을 기반으로 시스템 재사용성의 향상과 유지보수의 용이함을 위해 계층적 패턴[12, 13]을 사용하여 (그림 4)와 같이 3계층 표현 방식 아키텍처로 설계하였다. 3계층 표현 방식(three-layer expression formula)의 아키텍처는 보다 체계적이고 효율적으로 관계를 가질 수 있도록 하는 3개의 계층으로 구성된 표현 형태를 가지는 아키텍처를 의미하는 것이다[11]. 이것은 각기 기 구축된 데이터베이스의 독립성을 최대한 유지하면서 이질적이고 분산되어 있는 데이터베이스를 마치 하나의 데이터베이스 처럼 투명성 있게 사용할 수 있는 통합 아키텍처이다.



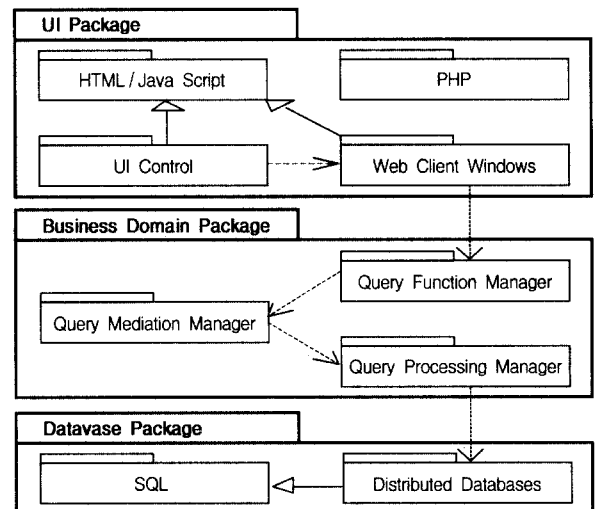
(그림 4) MDR 기반의 분산 정보 통합 시스템의 3계층 아키텍처

사용자 인터페이스 레이어(UI Layer)는 클라이언트들을 통해 사용자 질의에 대한 입출력을 담당하는 계층이고, 분산 정보 통합 시스템 레이어(DIIS Layer)는 세 개의 핵심 관리기인 질의 함수 관리기(QFM : Query Function Manager), 질의 중개 관리기(QMM : Query Mediation Manager), 질의 처리 관리기(QPM : Query Processing Manager)를 통해 사용자 질의 함수에 대한 분석, 변환, 수행, 통합 등의 시스템

의 핵심적인 기능을 수행하는 계층이며, 데이터베이스 인터페이스 레이어(DB Interface Layer)는 분산된 이기종의 로컬 데이터베이스 접근에 대한 처리를 담당하는 계층이다.

이러한 3계층 아키텍처의 핵심 기능은 다음과 같다. ① 질의 함수 관리기와 질의 함수 저장소(QFR)를 통하여 질의 캐싱(Query Caching) 기능과 질의 재사용성(Query Reusability)을 제공한다. 즉, 비용 기반의 질의 최적화 기법을 사용하여 자주 사용하는 질의나 질의 분할이 자주 이루어지는 질의인 경우에는 질의 함수 형태로 저장하여 사용함으로써 질의 처리 속도와 질의 재사용성을 향상시킨다. ② 분산 데이터베이스 시스템의 원활한 정보 공유와 교환을 위해 메타데이터 레지스트리(MDR)와 스키마 레파지토리(SCR)를 이용함으로써 이질성을 해결한다.

3계층 아키텍처를 기반으로 분산 정보 통합 시스템을 각각의 서브 시스템들로 구성하는데 필요한 패키지들을 추출한 후 (그림 5)와 같이 패키지 다이어그램으로 모델링하였다. 분산 정보 통합 시스템의 UI Layer, DIIS Layer, DB Interface Layer는 각각 UI Package, Business Domain Package, Database Package로 표현되었다. UI Package는 사용자와 시스템 간의 인터페이스를 하는데 있어서 중요한 역할을 담당한다. Business Domain Package는 세 개의 핵심 컴포넌트인 질의 함수 관리기 컴포넌트, 질의 중개 관리기 컴포넌트, 질의 처리 관리기 컴포넌트로 구성되며, 시스템의 목표를 달성하는데 필요한 비즈니스 도메인 기능들을 수행한다. Database Package는 시스템이 다양한 종류의 이기종의 분산된 데이터베이스에 접근할 수 있도록 해 준다.



(그림 5) 분산 정보 통합 시스템의 패키지 다이어그램

### 3.2.1 질의 함수 관리기(QFM : Query Function Manager)

질의 함수 관리기는 웹 기반의 사용자 질의에 대한 질의 함수가 질의 함수 저장소에 존재하는지를 체크하여, 질의 함

수가 존재하는 경우에는 질의 함수 저장소로부터 질의 함수를 불러오고 QFTag에 TRUE를 할당한 뒤 불러온 질의 함수와 QFTag를 질의 중개 관리자에게 넘겨주며, 질의 함수가 존재하지 않는 경우에는 새로운 질의 함수를 동적으로 생성하고 QFTag에 FALSE를 할당한 뒤 생성된 질의 함수와 QFTag를 질의 중개 관리자에게 넘겨주는 역할을 한다.

질의 함수 관리기는 질의 캐싱 기능과 질의 재사용성 기능을 제공한다. 즉, 질의 함수 관리기는 웹 기반의 사용자 질의에 대한 질의 함수가 존재하는 경우, 질의 함수 저장소로부터 질의 함수를 불러온 후 사용하므로 질의 분석 및 변환 과정을 거치지 않으므로 질의 처리 속도를 향상시킬 수 있다. 사용자 질의에 대한 질의 함수가 존재하지 않는 경우에는 동적으로 질의 함수를 생성한 후 질의 중개 관리기에서 메타데이터 레지스트리와 스키마 레퍼지토리를 참조하여 질의 함수를 분석하고 변환한 뒤, 비용 기반의 질의 최적화 기법을 사용하여 자주 질의가 사용되거나 질의 분할이 자주 이루어진다고 판단되면, 그 질의 함수를 질의 함수 저장소에 저장해 놓음으로써 똑같은 패턴의 질의에 대해 질의 캐싱 기능과 질의 재사용성의 기능을 제공한다.

질의 함수의 포맷은 (그림 6)에서 보여주고 있고, 관리자가 서울 DB인 수주 서비스 카테고리에서 수주 기간을 검색인자로 하는 질의 함수의 예제는 (그림 7)에서 보여주고 있다.

```
function (관리자, 검색서비스명, 검색인자 1, 검색인자 2, ...,
    검색인자 n) {
    SELECT 검색항목 1, 검색항목 2, ..., 검색항목 n
    FROM 로컬테이블 1, 로컬테이블 2, ..., 로컬테이블 0
    WHERE 검색인자i = 조건1
        :
        AND 검색인자j = 조건p
        AND 로컬 DBi = 로컬 DBj
}
```

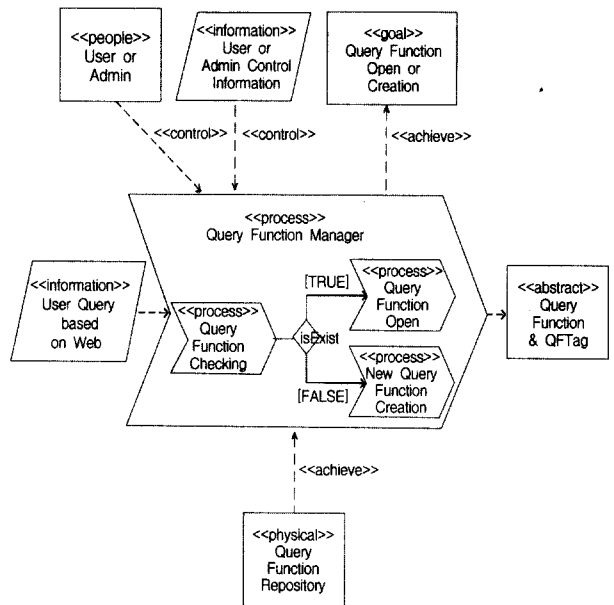
(그림 6) 질의 함수 포맷

```
function (서울, 수주서비스, 수주기간) {
    SELECT 수주일자, 수주번호, 주문처명, 품목명, 주문량, 주문
    비용, 납기일
    FROM 수주테이블, 납기테이블, 품목테이블
    WHERE 수주일자 BETWEEN '수주시작일' AND '수주최종일'
    AND 수주테이블 primary Key = 납기테이블 primary Key
    AND 수주테이블 primary Key = 품목테이블 primary Key
}
```

(그림 7) 질의 함수 예제

질의 함수 관리기의 비즈니스 프로세스간의 협력 관계를 보여주기 위해 (그림 8)과 같은 EPDM 방법론의 프로세스 다이어그램을 통해 표현하였다. 프로세스 다이어그램은 각 컴포넌트에 대한 프로세스와 리소스 객체들을 추출한 뒤 그

들간의 상호 인터페이스를 도출함으로써 그려진다. Query Function Manager 프로세스는 세 개의 서브 프로세스인 Query Function Checking, Query Function Open, New Query Function Creation으로 구성된다. 이 프로세스의 목적은 질의 함수를 불러오거나 생성하는 것이다. 입력으로는 웹 기반의 사용자 질의가 들어오며, 출력으로는 Query Function과 QFTag의 값이 나간다. 일반 사용자(User)인지 관리자(Admin)인지를 나타내는 People 리소스와 그들에 대한 제어 정보를 나타내는 Information 리소스가 필요하며, 물리적인 리소스로는 Query Function Repository가 필요하다.

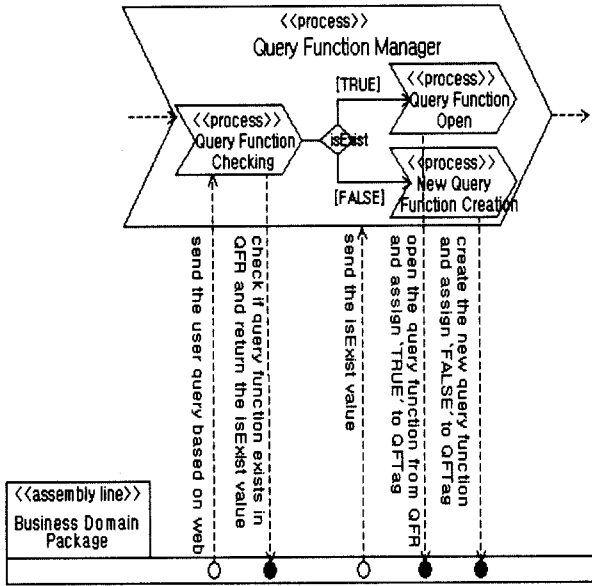


(그림 8) 질의 함수 관리기의 프로세스 다이어그램

Query Function Manager 프로세스는 프로세스 내에서 분기문을 사용함으로써 사용자 질의에 해당하는 질의 함수가 Query Function Repository에 존재하는지를 체크한다. 질의 함수가 존재하는 경우에는 Query Function Repository로부터 질의 함수를 저장하고 저장된 질의 함수와 QFTag를 질의 중개 관리자에게 넘겨준다. 질의 함수가 존재하지 않는 경우에는 새로운 질의 함수를 생성한 뒤, 질의 함수가 자주 사용되거나 질의 분할이 자주 이루어지는지를 판단하여, 그런 경우에는 생성된 질의 함수를 Query Function Repository에 저장하는 절차를 한번 더 거친 후, 생성된 질의 함수와 QFTag를 질의 중개 관리자에게 넘겨준다.

질의 함수 관리기의 비즈니스 프로세스 Query Function Manager와 Business Domain Package간의 상호작용을 보여주기 위해 (그림 9)와 같은 EPDM 방법론의 어셈블리 라인 다이어그램을 통해 표현하였다. 어셈블리 라인 다이어그램은 비즈니스 프로세스와 그에 수반되는 객체 사이의 관

계에 초점을 두고 있다. 이 다이어그램의 목적은 다이어그램 윗 부분에 표현되는 프로세스들이 어떻게 객체들을 읽고 쓰는지를 보여주는 것이다[16]. 어셈블리 라인상에 있는 *Business Domain Package*가 *Query Function Manager* 프로세스와 어떻게 상호 작용하는지를 '○'로 표시된 입력과 '●'로 표시된 출력을 통하여 상세히 표현하고 있다.



(그림 9) 질의 함수 관리기의 어셈블리 라인 다이어그램

### 3.2.2 질의 중개 관리기(QMM : Query Mediation Manager)

질의 중개 관리기는 질의 함수가 존재하는지의 여부를 나타내는 QFTag의 값을 체크하여, 질의 함수가 존재하는 경우에는 저장된 질의 함수를 질의 처리 관리기에 바로 넘겨줌으로써 질의 처리 속도를 향상시켜 주며, 질의 함수가 존재하지 않는 경우에는 저장된 질의 함수를 메타데이터 레지스트리와 스키마 레파지토리를 이용하여 질의를 분석(질의 분석기 : Query Analyzer)한 후 각각의 로컬 DB에 접근할 수 있는 형태로 변환(질의 변환기 : Query Transformer)한 뒤, 질의 함수가 자주 사용되거나 질의 분할이 자주 이루어진다고 판단되면 변환된 질의 함수를 질의 함수 저장소에 저장하는 절차를 한번 더 거친 후, 변환된 질의 함수를 질의 처리 관리기에 넘겨주는 역할을 수행한다.

질의 함수 관리기는 분산 데이터베이스 시스템의 원활한 정보 공유와 교환을 위해 메타데이터 레지스트리와 스키마 레파지토리를 이용함으로써 스키마 이질성과 데이터 이질성 문제를 해결한다. (그림 10)은 메타데이터 레지스트리와 스키마 레파지토리간의 매핑 관계를 보여준다. 먼저 위쪽은 '수주'에 대한 메타데이터 레지스트리 설계서의 일부를 보여준다. 실제계의 생각, 추상, 사물의 집합을 나타내는 OBJECT는 '수주'가 되고, 객체 부류 내의 모든 구성원을 특징짓는

성격인 PROPERTY는 '코드', '이름', '수량' 등이 되며, 값, 영역, 데이터 형, 측정 단위 등의 표현 형태를 나타내는 REPRESENTATION은 'TYPE', 'LENGTH', 'UNIQUE'로 분류하여 표시된다. 검색인자로 쓰이는 데이터 요소(DATA ELEMENT)는 유일한 MDR ID 값을 가지며, 사용자 질의에 대한 통합된 정보는 BIGO의 표현 형태를 통하여 통일된 표현 방식으로 보여지게 되는 것이다. 다음으로 아래쪽은 스키마 레파지토리 설계서의 일부를 보여준다. 여기에는 실제 분산되어 있는 스키마들의 정보가 포함되어 있음을 알 수 있다. SCR\_DB\_NAME은 관리지사 DB를 나타내고, SCR\_TABLE\_NAME은 각 관리지사의 테이블명을 나타내며, SCR\_COLUMN\_NAME은 각 테이블의 컬럼명을 나타낸다. 각 컬럼명은 유일한 스키마 ID(SCR\_SCHEMA\_ID)를 가지며, 스키마 ID는 MDR ID와 매핑되어 있다.

Metadata Registry 설계서		버전	2.0	작성자	김중환	
		페이지	4	작성일	2002.05.01	
OBJECT CLASS		수주				
MDR ID	PROPERTY	REPRESENTATION			DATA ELEMENT	BIGO
		TYPE	LENGTH	UNIQUE		
MDR0401	코드	CHAR	10	YES	수주코드	ORDE-XXX-XXXX
MDR0402	이름	CHAR	20	YES	총괄명	데이터 이질성 해결
MDR0403	스키마 이질성 해결		40	YES	담당처명	
MDR0404			12	NO	수량	개
MDR0405	단기	NUMBER	12	NO	수주금액	원
MDR0406	일자	CHAR	10	NO	수주일자	XXXX-XX-XX
MDR0407	시간	CHAR	8	NO	수주시간	AM/PM XX:XX

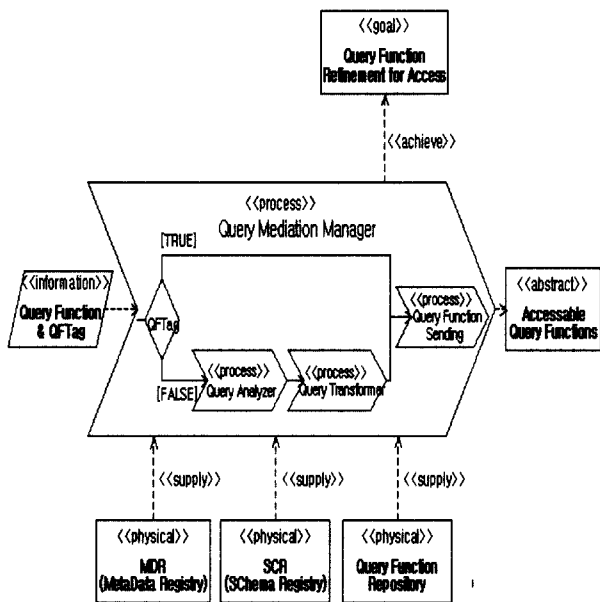
SCR_SCHEMA_ID	SCR_DB_NAME	SCR_TABLE_NAME	SCR_COLUMN_NAME	MDR_ID
SCR0029	서울	ORDER_LIST	ORD_ID	MDR0401
SCR0035	서울	ORDER_LIST	ORD_DATE	MDR0406
SCR0103	부산	ORDER_TABLE	O_CODE	MDR0401
SCR0108	부산	ORDER_TABLE	O_DATE	MDR0406
SCR0167	울산	ORDERING	ORDER_ID	MDR0401
SCR0173	울산	ORDERING	ORDER_DATE	MDR0406

(그림 10) 메타데이터 레지스트리와 스키마 레파지토리의 매핑

① 검색인자로 '수주코드'가 들어오면 ② '수주코드'에 해당하는 MDR ID 'MDR 0401'를 찾은 후 ③ 이것을 가지고 스키마 레파지토리의 MDR\_ID로 이동한 뒤 ④ 각 MDR\_ID에 대한 SCR\_SCHEMA\_ID 'SCR 0029', 'SCR 0103', 'SCR 0167'에 각각 매핑됨으로써 스키마 이질성을 해결한다. 스키마 ID 'SCR 0029'는 서울 DB의 'ORDER\_LIST' 테이블에서 'ORD\_ID'라는 컬럼으로 들어가 있고, 'SCR 0103'은 부산 DB의 'ORDER\_TABLE' 테이블의 'O\_CODE'라는 컬럼으로 들어가 있으며, 'SCR 0167'은 울산 DB의 'ORDERING' 테이블의 'ORDER\_ID'라는 컬럼으로 들어가 있다. 또한 ⑤ '수주코드' 포맷을 메타데이터 레지스트리 설계서의 'BIGO' 부분에 명시해 줌으로써 데이터 이질성을 해결한다. 따라서 메타데이터 레지스트리와 스키마 레파지토리를 이용함으로써

서 스키마 이질성 및 데이터 이질성을 해결할 수 있으며, 이로써 분산 데이터베이스 시스템의 원활한 정보 공유와 교환이 가능하게 되는 것이다.

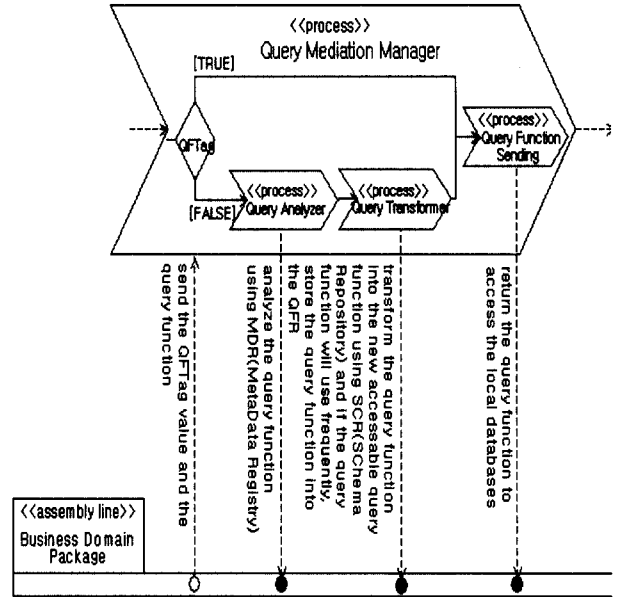
질의 중개 관리기의 비즈니스 프로세스 간의 협력 관계를 보여주기 위해 (그림 11)과 같은 EPEM 방법론의 프로세스 다이어그램을 통해 표현하였다. *Query Mediation Manager* 프로세스는 세 개의 서브 프로세스인 *Query Analyzer*, *Query Transformer*, *Query Function Sending*으로 구성된다. 이 프로세스의 목적은 로컬 데이터베이스에 접근 가능한 질의 함수를 만드는 데 있다. 입력으로는 *Query Function*과 *QFTag*가 들어오며, 출력으로는 로컬 DB에 접근 가능한 질의 함수가 나가게 된다. 물리적 리소스로는 *MDR*과 *SCR*이 필요하다. *Query Mediation Manager* 프로세스는 프로세스 내에서 분기문을 사용함으로써 *QFTag*의 값을 체크하여 사용자 질의에 대한 질의 함수가 존재하는 경우 저장된 질의 함수를 바로 질의 처리 관리기로 넘겨주고, 존재하지 않는 경우에는 생성된 질의 함수를 *MDR*과 *SCR*을 참조하여 분석 및 변환한 후, 질의 함수가 자주 사용되거나 질의 분할이 자주 이루어진다고 판단되면 변환된 질의 함수를 질의 함수 저장소에 저장하는 절차를 한번 더 거친 후, 변환된 질의 함수를 질의 처리 관리기에게 넘겨주는 일을 하게 된다.



(그림 11) 질의 중개 관리기의 프로세스 다이어그램

질의 중개 관리기의 비즈니스 프로세스와 패키지 간의 상호작용을 보여주기 위해 (그림 12)와 같은 EPEM 방법론의 어셈블리 라인 다이어그램을 통해 표현하였다. 어셈블리 라인 상에 있는 *Business Domain Package*가 *Query Mediation Manager* 프로세스와 어떻게 상호 작용하는지를 '○'

로 표시된 입력과 '●'로 표시된 출력을 통하여 상세히 보여주고 있다.

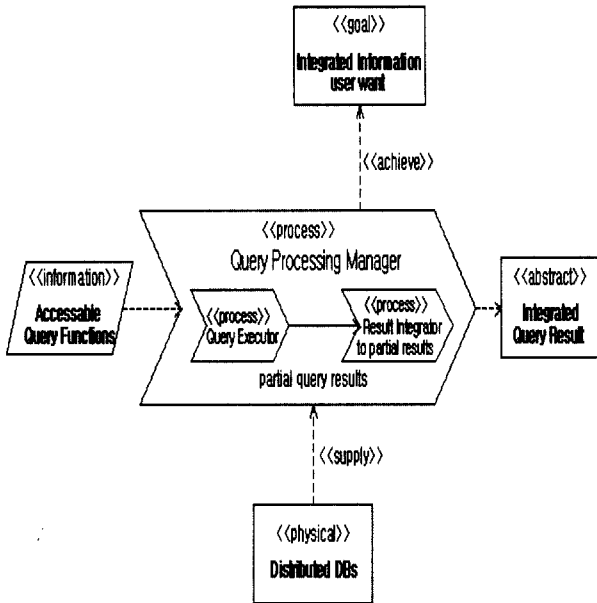


(그림 12) 질의 중개 관리기의 어셈블리 라인 다이어그램

### 3.2.3 질의 처리 관리기(QPM : Query Processing Manager)

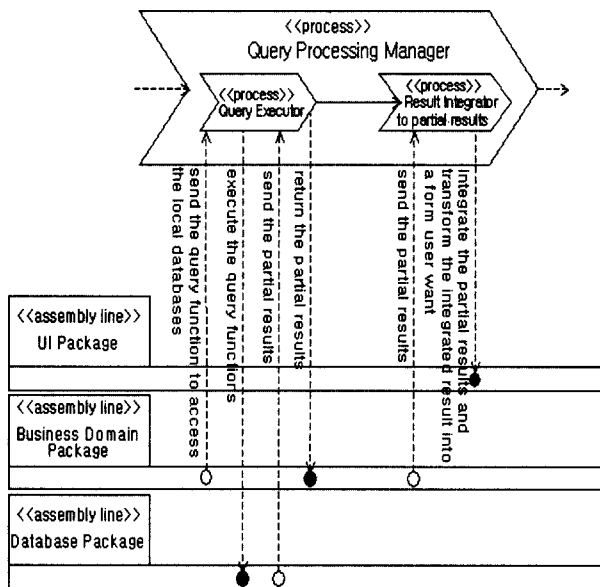
질의 처리 관리기는 질의 수행기(Query Executor)를 통하여 질의 함수를 각각의 로컬 데이터베이스에 접근할 수 있도록 wrapper를 이용하여 변환한 뒤 각각의 질의를 로컬 데이터베이스에서 수행한다. 수행된 후 얻어진 각 부분 질의 결과들은 결과 통합기(Result Integrator)에 넘겨지게 되며, 결과 통합기는 이러한 부분 질의들을 통합한 뒤 메타데이터 레지스트리 설계서에 나와 있는 포맷에 맞춰 가공한 뒤, 사용자 인터페이스에게 넘겨주는 역할을 한다.

질의 처리 관리기의 비즈니스 프로세스 간의 협력 관계를 보여주기 위해 (그림 13)과 같은 EPEM 방법론의 프로세스 다이어그램을 통해 표현하였다. *Query Processing Manager* 프로세스는 두 개의 서브 프로세스인 *Query Executor*, *Result Integrator*로 구성된다. 이 프로세스의 목적은 사용자가 원하는 통합된 정보를 만드는 데 있다. 입력으로는 로컬 데이터베이스에 접근 가능한 질의 함수가 들어오며, 출력으로는 사용자가 원하는 통합된 정보가 나가게 된다. 물리적 리소스로는 분산된 데이터베이스들이 필요하다. *Query Processing Manager* 프로세스는 로컬 데이터베이스에 접근 가능한 질의 함수들을 각각의 데이터베이스에 접근함으로써 결과들을 얻어오며, 이러한 부분 결과들을 통합하여 하나의 통합된 포맷으로 가공한 뒤, 사용자 인터페이스에게 넘겨주는 일을 수행하게 된다.



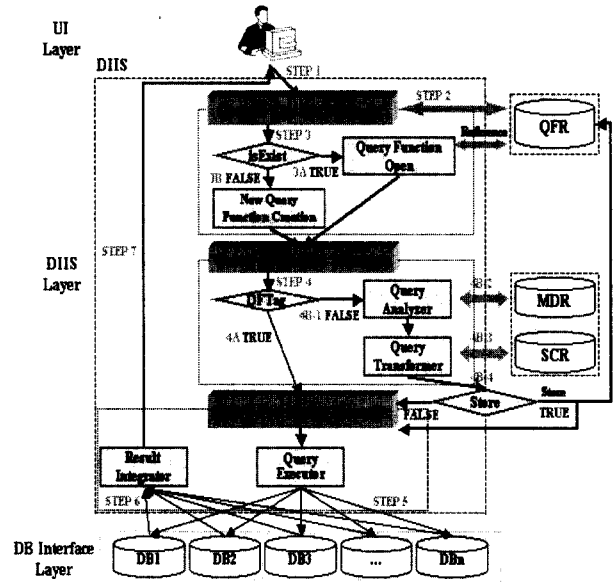
(그림 13) 질의 처리 관리기의 프로세스 다이어그램

질의 처리 관리기의 비즈니스 프로세스와 패키지 간의 상호작용을 보여 주기 위해 (그림 14)와 같은 EPDM 방법론의 어셈블리 라인 다이어그램을 통해 표현하였다. 어셈블리 라인 상에 있는 UI Package, Business Domain Package, Database Package가 Query Processing Manager 프로세스와 어떻게 상호 작용하는지를 '○'로 표시된 입력과 '●'로 표시된 출력을 통하여 상세히 보여주고 있다.



(그림 14) 질의 처리 관리기의 어셈블리 라인 다이어그램

3.3 분산 정보 통합 시스템의 질의 처리 절차와 통합 설계  
 분산 정보 통합 시스템의 질의 처리 절차는 (그림 15)와 같다.



(그림 15) 분산 정보 통합 시스템의 질의 처리 절차

- STEP 1 : 웹 기반의 사용자 질의를 입력받아 Query Function Manager에 넘겨준다.
- STEP 2 : 질의 함수가 QFR에 존재하는지를 체크하여 is-Exist에 TRUE나 FALSE의 값을 할당한다.
- STEP 3 : 질의 함수가 존재하는지의 여부를 나타내는 is-Exist의 값을 조사하여,
  - 3A : isExist가 TRUE인 경우, QFR로부터 질의 함수를 불러오고 QFTag에 TRUE를 할당한 뒤, 불러온 질의 함수와 QFTag를 Query Mediation Manager에게 넘겨준다.
  - 3B : isExist가 FALSE인 경우, 질의 함수를 동적으로 생성한 후, QFTag에 FALSE를 할당하고, 생성된 질의 함수와 QFTag를 Query Mediation Manager에게 넘겨준다.
- STEP 4 : 질의 함수가 존재하는지의 여부를 나타내는 QFTag의 값을 조사하여,
  - 4A : QFTag가 TRUE인 경우, 불러온 질의 함수를 Query Processing Manager에 바로 넘겨준다.
  - 4B-1 : QFTag가 FALSE인 경우, 새로 생성된 질의 함수를 Query Analyzer에게 넘겨준다.
  - 4B-2 : Query Analyzer는 MDR과 SCR을 이용하여 질의 함수를 분석하고 분석한 질의 함수를 Query Transformer에게 넘겨준다.
  - 4B-3 : Query Transformer는 질의 함수를 로컬 데이터베이스에 접근할 수 있는 새로운 질의 함수로 변환한다.
  - 4B-4 : 변환된 새로운 질의 함수가 자주 질의가 사용되거나 자주 질의 분할이 일어난다면, 변환된



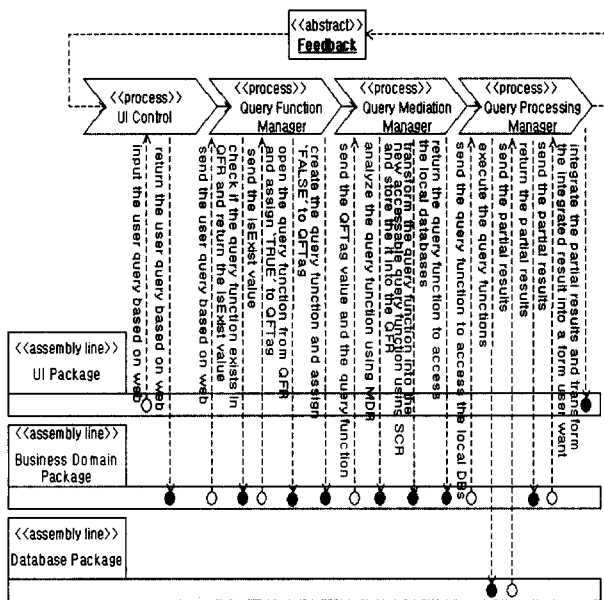
새로운 질의 함수를 QFR에 저장하는 절차를 한 번 더 거친 후, 변환된 질의 함수를 Query Processing Manager에 넘겨준다.

**STEP 5 :** Query Executor에서는 질의 함수가 각각의 로컬 DB에 접근할 수 있도록 wrapper를 통하여 변환한 뒤 각각 수행하고, 수행된 부분 질의 결과들을 Result Integrator에 넘겨준다.

**STEP 6 :** Result Integrator에서는 부분 질의 결과들을 통합한 뒤 메타데이터 레지스트리 설계서에 나와 있는 통일된 포맷으로 가공한 후 User Interface에게 넘겨준다.

**STEP 7 :** User Interface는 통합된 정보를 넘겨받아 화면에 보여준다.

분산 정보 통합 시스템의 통합 설계는 어셈블리 라인 상에서 프로세스와 패키지들이 서로 어떻게 상호 작용하는지를 전체적으로 표현하기 위하여, 기본적인 프로세스 구조패턴과 프로세스 정보패턴[14]에 기반하여 (그림 16)과 같이 통합 어셈블리 라인 다이어그램을 통해 표현하였다. 즉, 각각의 프로세스(UI Control, Query Function Manager, Query Mediation Manager, Query Processing Manager)가 각각의 패키지(UI Package, Business Domain Package, Database Package)들과 어떻게 입출력되는지를 보여주고 있다. 예를 들어, UI Package가 사용자 질의를 요청하면, UI Control 프로세스는 사용자 질의를 Business Domain Package로 리턴해 준다. 또한 이 다이어그램에서는 사용자가 로그아웃을 통하여 종료하지 않는 한 계속 루프를 도는 형태이기 때문에 Feedback을 사용하였다.



(그림 16) 분산 정보 통합 시스템의 통합 어셈블리 라인 다이어그램

#### 4. 분산 정보 통합 시스템의 구현

본 장에서는 이러한 설계를 바탕으로 앞에서 제안한 분산 정보 통합 시스템의 구체적인 한 예로서, 공급망 관리 도메인의 ‘수주 관리’ 부분에 적용하여 구현하였다. 구현된 화면을 통하여 질의 캐싱을 통한 질의 처리 속도 향상과 메타데이터 레지스트리와 스키마 레퍼지토리를 이용한 이질성 해결을 설명한다.

##### 4.1 공급망 관리(SCM : Supply Chain Management)

공급망 관리란 공급자로부터 소비자를 연결하는 개발, 조달, 제조, 배송, 판매라는 일련의 업무를 효율적으로 관리하기 위한 경영전략으로, 공급업체, 제조업체, 유통업체, 소매업체, 소비자가 상호 연계되어 있으며, 공급망 전체의 업무 프로세스 흐름을 한개의 통합된 비즈니스 프로세스로 간주하고 기업과 조직의 벽을 넘는 비즈니스 프로세스의 전체 최적화를 추구하여 결과적으로 양질의 상품 및 서비스를 소비자에게 제공함으로써 소비자 가치를 극대화하는데 목적이 있다[15]. 세계적으로 선도적 위치에 있는 공급업체, 제조업체, 유통업체, 소매업체들은 이러한 목적을 달성하기 위하여 그들의 거래선들과 협력함으로써 그 이익을 훨씬 더 극대화하였으며, 공급망 관리를 추진중에 있는 대표적인 업체로는 P&G, Wal-Mart, Unilever, Kellogg’s, Coca Cola 등이 있다[16].

##### 4.2 의사 코드로 표현한 시스템의 전체 흐름

제안된 분산 정보 통합 시스템의 전체적인 흐름을 (그림 17)과 같이 의사 코드를 사용하여 표현하였다. 의사 코드(pseudo code)란 일반적으로 쓰기 쉽고 읽기 쉬우며 알고리즘의 다섯 가지 조건인 입력, 출력, 명확성, 유한성, 효과성을 만족시키는 표현 방법이다. 이것은 프로그래밍 언어에 가까우며 특정한 프로그래밍 언어에 제약을 받지 않는다[17].

```

Algorithm overall_flow /* Overall Flow of the DIIS */
Input : user query based on web
Output : integrated information or error message
{
/* UI : User Interface based on web */
input ID and PW ;/* user or admin authentication */
if (ID and PW is not correct) return error message ;
else
go to the main screen ;
input a user query based on web ;
send the user query to QFM ;

/* QFM : Query Function Manager of the DIIS */
if(query function exists in QFR) { /* isExist = 참 */
open the query function from QFR ;
assign TRUE to QFTag ;
} else { /* isExist = 거짓 */
create a query function ;
assign FALSE to QFTag ;
}
send QFTag and query function to QMM ;
}
    
```

```

/* QMM : Query Mediation Manager of the DIIS */
if(QFTag is TRUE) {
    send the query function to QPM ; /* 질의 캐싱 */
} else if(QFTag is FALSE) {
    call the query analyzer ; /* 질의 분석 */
    call the query transformer ; /* 질의 변환 */
    if (query function will use frequently) { /* 질의 저장 */
        store the new transformed query function into QFR
    }
    send the transformed query function to QPM ;
}

/* QPM : Query Processing Manager of the DIIS */
call the Query Executor ; /* 질의 수행 */
call the Result Integrator ; /* 질의 통합 */
send the integrated information to the UI ;

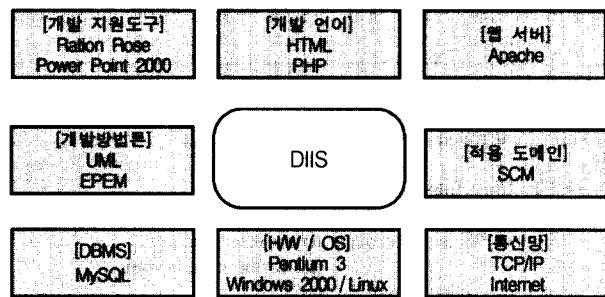
/* UI : User Interface based on web */
display the integrated information ;
}
    
```

(그림 17) 의사 코드로 표현한 분산 정보 통합 시스템의 전체 흐름

4.3 분산 정보 통합 시스템의 구현

4.3.1 시스템 개발 및 운영 환경

구현된 분산 정보 통합 시스템의 개발 및 운영 환경은 (그림 18)과 같다. 분산 정보 통합 시스템의 기능성과 흐름을 표현하기 위하여 EPDM 개발 방법론을 이용하여 설계하였다. 시스템 구현 도메인은 공급망 관리에서 수주 관리 부분이며, 개발 언어로는 PHP와 HTML을, 웹 서버로는 Apache를, DBMS로는 MySQL을, OS로는 Linux와 Windows 2000을 사용하여 인터넷을 통해서 웹 상에서 동작할 수 있도록 구현하였다. 시스템 개발시, 서로 다른 스키마를 가지는 관계형 환경을 전제로 하였기 때문에 래퍼의 기능은 구현되지 않았다.

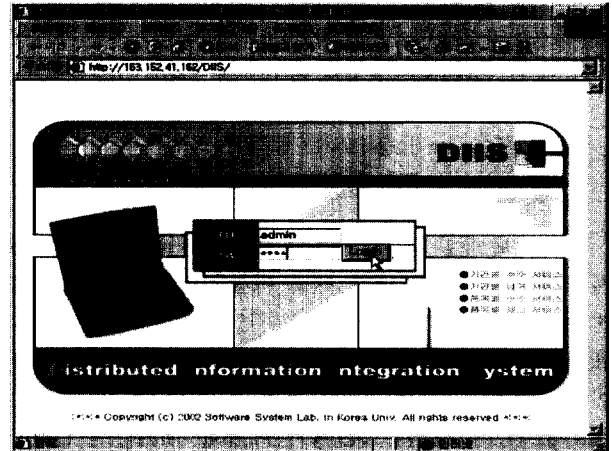


(그림 18) 분산 정보 통합 시스템의 개발 및 운영 환경

4.3.2 시스템 초기 화면

질의 함수 저장소, 메타데이터 레지스트리, 스키마 레퍼지토리를 기반으로 하는 분산 정보 통합 시스템을 공급망 관리 도메인에 적용하여 구현하였다. (그림 19)는 분산 정보 통합 시스템의 초기 화면을 보여주고 있다. 초기 화면은 관리자(admin)와 일반 사용자(user) 인증을 하는 로그인 화면이다.

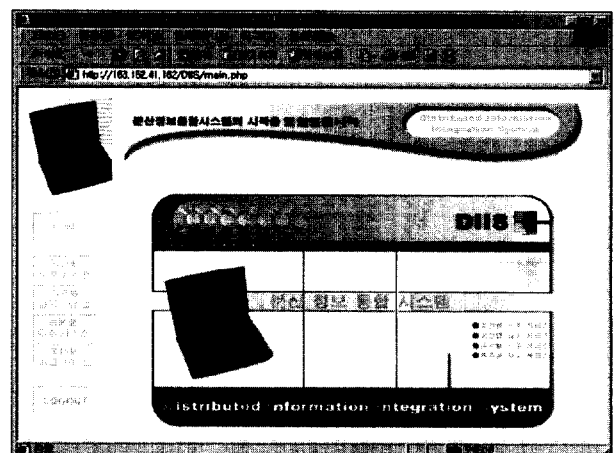
관리자와 일반 사용자에게 보여지는 메인 화면은 각각 다르며, 인증이 정상적으로 이루어진 경우 각각의 메인 화면으로 이동하고, 그렇지 않은 경우에는 오류 메시지 창이 띄운 후 초기 화면으로 이동하게 된다. 관리자로 로그인한 경우에 나타나는 메인 화면에는 일반 사용자에게 보여지는 메인 화면에 메타데이터 레지스트리와 스키마 레퍼지토리를 관리할 수 있는 기능이 추가되어 있다.



(그림 19) 분산 정보 통합 시스템의 초기 화면

4.3.3 시스템 메인 화면

정상적인 인증 과정을 거치게 되면 (그림 20)과 같은 분산 정보 통합 시스템의 일반 사용자 메인 화면이 보여지게 된다. 여기서는 질의 캐싱과 질의 재사용성 및 이질성 문제를 다루기 위해 일반 사용자 화면만을 설명한다. 화면 왼쪽 부분은 고정된 부분으로 'HOME', '기간별 수주 서비스', '기간별 납기 서비스', '품목별 수주 서비스', '품목별 재고 서비스', 'LOGOUT'의 항목을 보여주고 있다. 화면 오른쪽 부분은 유동적인 부분으로 화면 왼쪽의 해당 아이콘을 누를 때마다 각각에 해당하는 화면이 보여지게 된다. 'LOGOUT'을 클릭하게 되면 초기 화면으로 이동하게 된다.

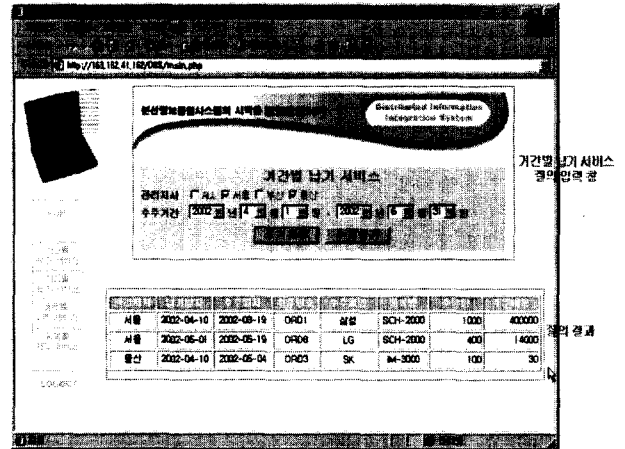


(그림 20) 분산 정보 통합 시스템의 메인 화면

4.3.4 기간별 수주 서비스 화면

메인 화면의 '기간별 수주 서비스' 아이콘을 선택하게 되면, (그림 21)의 화면 오른쪽 상단에 보여지는 것과 같은 '기간별 수주 서비스 질의 입력 창'만이 뜨게 된다. 관리자 사 DB는 '서울', '부산', '울산'의 세 곳에 있다고 가정한다. 'ALL'을 선택하게 되면 '서울', '부산', '울산'의 모든 DB를 검색하게 되고, 'ALL'을 선택하지 않고 각각을 선택하게 되면 각각의 DB의 내용만 검색하게 된다. 수주 기간을 콤보 박스 형태로 선택할 수 있게 하여 사용자 편리성을 도모하였다. 관리자를 'ALL'을 선택하고 수주 기간을 2002년 1월 1일부터 2002년 12월 31일까지 선택한 후 '수행하기' 아이콘을 클릭하게 되면, 지금까지 보이지 않던 (그림 21)의 화면 오른쪽 하단에 보여지는 것과 같은 '질의 결과 창'이 뜨면서 각 관리자 사 DB의 통합된 정보를 메타데이터 레지스트리 설계서를 참조하여 HTML 형태의 통일된 포맷으로 가공한 뒤 사용자에게 보여주게 된다. 기간별 수주 서비스에 대한 검색 명령이 처음이므로 function(서울, 수주 서비스, 수주 기간), function(부산, 수주 서비스, 수주 기간), function(울산, 수주 서비스, 수주 기간)의 질의 함수를 생성하고, 메타데이터 레지스트리와 스키마 레퍼지토리를 이용하여 질의를 분석 및 변환한 뒤, 질의가 자주 사용되거나 자주 질의 분할이 일어난다고 판단되면 질의 함수 저장소에 변환된 질의 함수를 저장하는 절차를 한번 더 거친 후에, 질의를 수행하고 통합하여 메타데이터 레지스트리 설계서의 통일된 포맷으로 보여 주기 때문에 질의 처리 속도와 스키마 이질성 및 데이터 이질성을 해결하는 것이다.

31일까지의 내용을 검색하라는 명령을 '수행하기' 아이콘을 클릭함으로써 주게 되면, 지금까지 보이지 않던 (그림 22)의 화면 오른쪽 하단에 보여지는 것과 같은 '질의 결과 창'이 뜨면서 통합된 결과를 사용자에게 보여주게 된다.

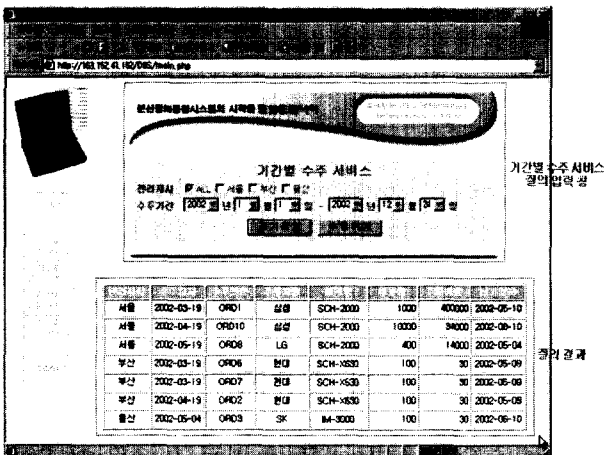


(그림 22) 기간별 납기 서비스 질의 입력 및 질의 결과

5. 비교 평가

본 장에서는 제시된 MDR 기반 분산 정보 통합 시스템에 대한 평가로서, 정보 통합 시스템의 관점에서 약한 결합 정보 통합 시스템, 연방 데이터베이스 정보 통합 시스템, 중개기 기반 정보 통합 시스템과, 비교 항목으로는 이질성 형태 (Types of Heterogeneity Addressed), 접근 방법(Access Methods), 메타데이터 사용(Use Metadata), ISO 11179 준수성(ISO11179 Compliance), 질의 캐싱(Query Caching), 질의 재사용성(Query Reusability), 접근 제한(Access Restrictions), 스키마 유연성(Schema Flexibility), 통합 강도(Tight vs. Loose Integration), 통합 절차(Bottom-up vs. Top-down) 등의 항목을 추출하여 <표 1>과 같이 비교 평가하였다[8, 18-20].

MDR 기반의 분산 정보 통합 시스템은 분산된 이질적인 형태의 각 데이터베이스를 단일의 인터페이스를 통하여 사용자가 각 데이터베이스에 대한 내부 구조를 몰라도 마치 단일의 시스템을 접근해서 정보를 얻는 것과 같은 투명성과 편리한 인터페이스를 제공하는 시스템이다. 구현된 시스템은 질의 언어(Query Language)를 사용하며, QFR, MDR, SCR을 통하여 이질성을 해결한다. ISO 11179를 따르는 메타데이터를 사용함으로써 레지스트리에 등록된 데이터 요소는 사용자나 개발자에게 전체 데이터베이스에 대한 통합적인 뷰를 제공하며 확장성을 지니게 된다. 이 시스템의 핵심적인 특징으로 질의 함수 관리기(QFM : Query Function Manager)와 질의 함수 저장소(QFR : Query Function Repository)를 통하여 질의 캐싱(Query Caching) 기능과 질의 재사용성(Query Reusability) 기능을 제공한다. 즉, 비



(그림 21) 기간별 수주 서비스 질의 입력 및 질의 결과

4.3.5 기간별 납기 서비스 화면

메인 화면의 '기간별 납기 서비스' 아이콘을 선택하게 되면, (그림 22)의 화면 오른쪽 상단에 보여지는 것과 같은 '기간별 납기 서비스 질의 입력 창'만이 뜨게 된다. 질의 입력 방법은 '기간별 수주 서비스'와 동일하다. '서울'과 '울산'에 해당하는 납기 기간이 2002년 4월 1일부터 2002년 6월

〈표 1〉 정보 통합 시스템 간의 비교 평가

비 교 항 목	약한 결합 정보 통합 시스템	연방 데이터베이스 정보 통합 시스템	증개기 기반 정보 통합 시스템	MDR 기반 분산 정보 통합 시스템
이질성 형태 (Types of Heterogeneity Addressed)	Only Technical and Language Heterogeneity	All, except query restriction heterogeneity	All	All
접근 방법 (Access Methods)	Query Language	Query Language	Any	Query Language
	Partially Yes	Yes		Yes
	No	No		No
	No	No		No
	No	No		Yes
접근 제한 (Access Restrictions)	No	No	Yes	Yes
스키마 유연성 (Schema Flexibility)	No	No	Partially Yes	No
통합 강도 (Tight vs. Loose Integration)	Loose	Tight	Tight	Tight
통합 절차 (Bottom-up vs. Top-down)	n.a.	Bottom-up	Top-down	Top-down

용 기반의 질의 최적화 기법을 사용하여 자주 사용하는 질의인 경우나 자주 질의 분할이 이루어지는 질의인 경우에는 질의 함수 형태로 저장하여 사용함으로써 질의 처리 속도와 질의 재사용성을 향상시켰다. 접근에 대한 제한이 있으며 스키마 구조가 변경되었을 때 유연하게 대처하는 능력이 떨어진다는 단점이 있다. 통합 강도로는 Tight Integration 방식을 사용하며, 사용자가 데이터베이스의 구조를 몰라도 원하는 통합된 정보를 얻을 수 있기 때문에 탑-다운(Top-down) 접근 방식을 사용한다.

6. 결 론

각기 기 구축된 데이터베이스의 독립성을 최대한 유지하면서, 이질적이고 분산되어 있는 데이터베이스를 마치 하나의 데이터베이스 처럼 투명성 있게 접근할 수 있는 통합 아키텍처인 분산 정보 통합 시스템을 시스템 재사용성의 향상과 유지보수의 용이함을 위해 계층적 패턴을 적용함으로써 3계층 표현 방식 아키텍처로 표현하였고, 이러한 시스템을 UML 방법론을 확장한 EPEM 방법론을 이용하여 설계한 뒤, 기업 집합 환경 하에서 운영되고 있는 공급망 관리 도메인에 적용하여 시스템을 구현하였다.

제안된 분산 정보 통합 시스템은 다음과 같은 특징을 지닌다. ① 질의 함수 관리기와 질의 함수 저장소를 통하여 질의 캐싱 기능과 질의 재사용성을 제공한다. 즉, 비용 기반의 질의 최적화 기법을 사용하여 자주 사용하는 질의나 질의 분할이 자주 이루어지는 질의인 경우에는 질의 함수 형태로 저장하여 사용함으로써 질의 처리 속도와 질의 재사용성을 향상시킨다. ② 분산 데이터베이스 시스템의 원활

한 정보 공유와 교환을 위해 메타데이터 레지스트리와 스키마 레퍼지토리를 이용함으로써 이질성을 해결한다. ③ 분산된 이질적인 형태의 각 데이터베이스를 단일의 인터페이스를 통하여 사용자가 각 데이터베이스에 대한 내부 구조를 몰라도 마치 단일의 시스템을 접근해서 정보를 얻는 것과 같은 투명성과 편리한 인터페이스를 제공한다. ④ ISO 11179를 따르는 메타데이터를 사용함으로써 레지스트리에 등록된 데이터 요소는 사용자나 개발자에게 전체 데이터베이스에 대한 통합적인 뷰를 제공한다.

향후 연구 과제로는, ① 분산된 데이터베이스의 스키마 구조가 변경되었을 때 유연하게 대처할 수 있는 스키마 유연성을 스키마 레지스트리 관리기를 통하여 해결하는 것이 필요하고 ② 이기종 데이터베이스의 통합을 위하여 랩퍼 기능의 구현을 추가할 필요가 있으며 ③ 데이터베이스의 수가 많아질 경우 질의 캐싱에 대한 성능 향상 여부를 통계적 수치로 표현함으로써 시뮬레이션하는 것이 요구된다.

참 고 문 헌

[1] Hea-Sook Park, Jong-Hwan Kim, Doo-Kwon Baik, "Componet-based Modeling and Verification of an Information Retrieval System using EPEM," Proceedings of The 4th ICACT, pp.821-826, Feb., 2002.  
 [2] 류근호, "생물다양성 정보의 투명한 접근을 위한 데이터베이스 분산 통합 방안", 한림 심포지움, 한국과학기술한림원, 2002.  
 [3] Unified Modeling Language Resoure Center, available online at : <http://www.rational.com/uml/>, 2002.  
 [4] Hans-Erik Eriksson and Magnus Penker, "Business Mo-

deling with UML,” OMG Press, 2000.

[5] A. P. Sheth and J. A. Larson, “Federated Database Systems for managing Distributed Heterogeneous and Autonomous Database,” ACM Computing Surveys, Vol.22, No. 3, pp.183-236, 1990.

[6] E. Kilic, G. et al, “Experiences in using CORBA for a Multidatabase Implementation,” 6th International Conference in Database, 1995.

[7] W. Litwin and A. Abdellatif, “Multidatabase Interoperability,” 1986.

[8] A. P. Sheth, J. A. Larson, “Federated Database System for Managing Distributed, Heterogeneous, and Autonomous Databases,” ACM Computing Surveys, Vol.22, No.3, pp.183-236, Sept., 1990.

[9] 이정욱, “멀티데이터베이스 시스템의 정보공유를 위한 개념 기반 의미망”, 고려대학교 대학원 컴퓨터학과 박사학위논문, 2001.

[10] ISO, Metadata Registry, ISO-11179, 2000.

[11] 송치양, 임성빈, 백두권, 김철홍, “컴포넌트 유통환경을 위한 컴포넌트 메타데이터 레지스트리 구축 : C\_MDR”, 정보과학회지, 제7권 제6호, 한국정보과학회, pp.614-629, 2001.

[12] Gamma, Erich, Richard, Helm, Ralph Johnson, John Vlissides, “Design Patterns,” Addison-Wesley, 1995.

[13] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, “Pattern-Oriented Software Architecture - A System of Patterns,” John Wiley & Sons, Inc., 1996.

[14] Hans-Erik Eriksson and Magnus Penker, “Enterprise Java with UML,” John Wiley & Sons, Inc., 2000.

[15] 후쿠시마 요시아키, “SCM 경영혁명”, 21세기 북스, 1998.

[16] ECR Korea, available online at : <http://scm.eankorea.or.kr/>, 2002.

[17] 황종선, 윤영식, “C 언어로 설명한 알고리즘”, 정의사, 1996.

[18] S. Busse, R. D. Kutsche, U. Leser, H. Weber, “Federated Information Systems : concepts, terminology and architectures,” Technical Report Nr.99-9, TU Berlin, 1999.

[19] C. Batini, M. Lenzerini, S. B. Navathe, “A Comparative Analysis of Methodologies for Database Schema Integration,” ACM Computing Surveys, Vol.18, No.4, pp.323-364, 1986.

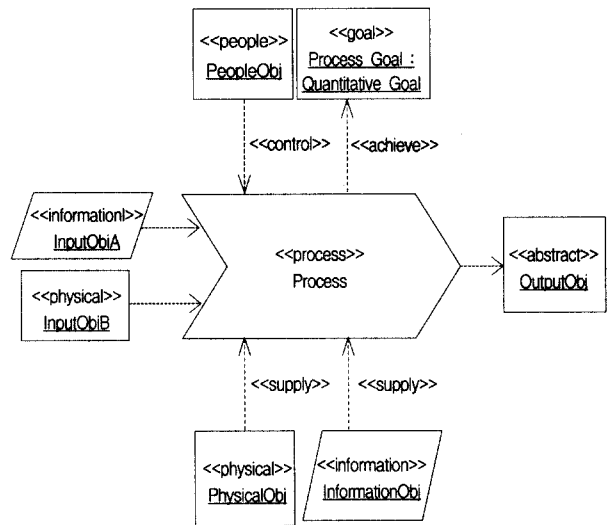
[20] G. Wiederhold, “Intelligent Integration of Information,” ACM SIGMOD Record, Vol.22, No.2, pp.434-437, Jun., 1993.

특징들을 더욱더 명확하게 식별하고 가시화하기 위해 UML 방법론에 비즈니스 프로세스 모델을 추가한 방법론이다.

EPEM은 기본적으로 UML의 표준 다이어그램인 유스케이스 다이어그램, 클래스 다이어그램, 액티비티 다이어그램을 사용하며, 액티비티 다이어그램의 특성화된 다이어그램인 프로세스 다이어그램(Process Diagram)과 어셈블리 라인 다이어그램(Assembly Line Diagram)을 사용한다. 프로세스 다이어그램은 비즈니스 프로세스 간의 협력관계를 보여주는 다이어그램이고, 어셈블리 라인 다이어그램은 비즈니스 프로세스와 패키지 간의 상호작용을 보여주는 다이어그램이다. EPEM은 '<< >>'로 표시되는 스테레오 타입이라 불리는 비즈니스 모델 요소들을 제공한다.

**A. 프로세스 모델링**

프로세스 모델링은 비즈니스 프로세스간의 협력 관계를 보여주는 프로세스 다이어그램을 가지고 표현된다. 프로세스 다이어그램은 프로세스간의 협력 관계를 다양한 스테레오 타입을 가지고 표현하는 액티비티 다이어그램의 일종으로, 프로세스의 목적 객체(Goal Object), 프로세스의 입력 객체(Input Object), 프로세스의 출력 객체(Output Object), 프로세스를 지원하는 객체(Supplying Object), 프로세스를 제어하는 객체(Controlling Object)로 구성된다. (그림 23)은 일반적인 프로세스 다이어그램을 보여준다.



(그림 23) 프로세스 모델링 : A generic Process Diagram

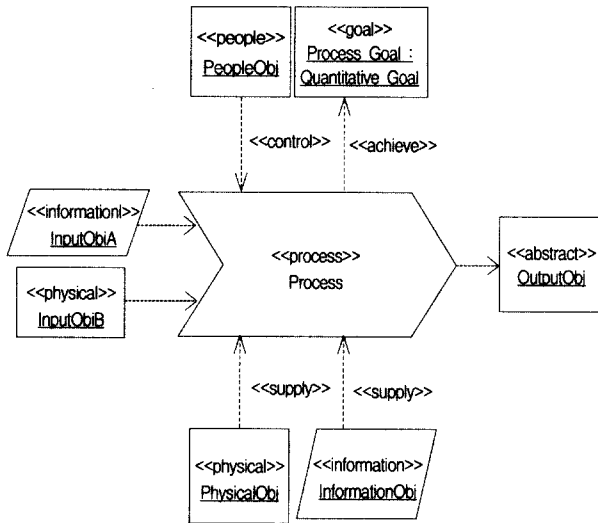
**<부 록> EPEM 방법론**

EPEM(Eriksson Penker Extension Model) 방법론은 UML 방법론을 확장한 새로운 비즈니스 프로세스 모델링 방법론의 하나이다. 즉, 소프트웨어 시스템을 개발할 때 하나의 모델링 표준인 UML 방법론이 복잡한 비즈니스 프로세스, 다양한 종류의 리소스들로 구성된 비즈니스 정보 시스템의 특징들을 완벽하게 표현할 수 없기 때문에, 그러한 시스템의

**B. 어셈블리 라인 모델링**

어셈블리 라인 모델링은 비즈니스 프로세스와 패키지 간의 상호작용을 보여주는 어셈블리 라인 다이어그램을 가지고 표현된다. 어셈블리 라인 다이어그램에서 어셈블리 라인 패키지는 하나의 긴 수평 직사각형으로 그려지며, '<<assembly line>>'이라는 스테레오타입을 갖는다. 이 다이어그램은 비즈니스 프로세스와 그에 수반되는 개체 사이의 관계에 초

점을 두며, 다이어그램 윗 쪽의 프로세스들이 다이어그램 아래 쪽의 객체들과 어떻게 읽고 쓰여지는지를 보여줌으로써, 어셈블리 라인 상에서 프로세스와 패키지들의 상호작용을 전체적으로 표현할 수 있다. (그림 24)는 일반적인 어셈블리 라인 다이어그램을 보여준다.



(그림 24) 어셈블리 라인 모델링 : A generic Assembly Line Diagram

**C. 표기법**

EPEM 방법론의 프로세스 다이어그램과 어셈블리 라인 다이어그램은 (그림 25)와 같은 표기법을 사용한다.

Process	Activity	Symbol	Description
Goal	Class	[[Goal]] Name	Dense to desired state, meaning that goals moderate actions leading to state changes in a desired direction.
Information	Class	[/Information/] Name	Information is a kind of resource. It is the knowledge increment brought about by a recording action in a message transfer.
Abstract	Class	[[Abstract]] Name	An abstract resource is an intangible asset. For example, mathematics, concepts, and so on.
People	Class	[[People]] Name	A physical resource; specifically, human beings.
Physical	Class	[/Physical/] Name	A physical resource, excluding people. For example, machines, documents, and so on.
Process Decision	Decision	◇	Decision point between two or more processes.
Process Flow	Control Flow	[[condition]]	A process control flow with a condition.
Resource Flow	Object Flow	---Name---	Object flow shows that an object is produced by one process and consumed by another process.
Process Control	Object Flow	<<control>>	Shows that a process is controlled by an object.
Goal Connection	Dependency	<<achieve>>	Allocates a goal to a process.
Process Supply	Object Flow	<<supply>>	Shows that a process is supplied by an object.
Assembly Line	Package	[[Assembly Line]]	The Assembly Lines synchronizes and supply processes in terms of objects.
Object-to-Assembly Line	Object	●	A delivered object from a process to the Assembly Line.
Object-from-Assembly Line	Object	○	An object that goes from the Assembly Line to a process.

(그림 25) EPEM 방법론 표기법

**김 종 환**



e-mail : angel96@software.korea.ac.kr  
 2001년 고려대학교 경제학과(학사)  
 2001년 고려대학교 컴퓨터학과(학사)  
 2001년~현재 고려대학교 대학원 컴퓨터학과 석사과정  
 관심분야 : 정보 통합, 메타데이터 레지스트리, 컴포넌트 기반 시스템, 정형 검증 등

**박 혜 숙**



e-mail : edpsphs@kic.ac.kr  
 1991년 고려대학교 산업공학과(학사)  
 1993년 고려대학교 대학원 산업공학과(석사)  
 1999년~현재 고려대학교 대학원 컴퓨터학과 박사과정

2002년~현재 경인여대 컴퓨터정보기술학부 전임강사  
 관심분야 : 정보 통합, 소프트웨어 개발 방법론, 전자 상거래 등

**문 창 주**



e-mail : mcj@software.korea.ac.kr  
 1997년 고려대학교 컴퓨터학과(학사)  
 1999년 고려대학교 대학원 컴퓨터학과(석사)  
 1999년~현재 고려대학교 대학원 컴퓨터학과 박사과정

관심분야 : 정보 통합, 컴포넌트 기반 시스템, 정보 보호

**백 두 권**



e-mail : baik@software.korea.ac.kr  
 1974년 고려대학교 수학과(학사)  
 1977년 고려대학교 대학원 산업공학과(석사)  
 1983년 Wayne State Univ.(전산학석사)  
 1985년 Wayne State Univ.(전산학 박사)

1986년~현재 고려대학교 컴퓨터학과 교수  
 1989년~현재 한국정보과학회 이사/평의원  
 1991년~현재 한국시물레이션학회 이사/부회장/감사  
 1991년~현재 ISO/IEC JTC1/SC32 국내위원회 위원장  
 2002년~현재 고려대학교 정보통신대학 학장  
 관심분야 : 데이터베이스, 소프트웨어 공학, 시물레이션, 메타데이터, 정보 통합, 정보 보호 등