분산환경에서 혼용 뷰 관리기법을 채택한 이질적인 멀티데이타베이스 상호운용 모델 설계

이 승 용[†] · 박 재 복[††] · 김 명 희[†††] · 주 수 종[††††]

요 약

본 논문에서는 이질 환경의 지역 데이터베이스 시스템(Local DataBase System: LDBS)을 분산 시스템 형태로 통합하고 전역 사용자에게 빠른 질의 처리를 제공하는 멀티 데이터베이스 관리 시스템(Multi-DataBase Management System: MDBMS)을 제안한다. 이를 위해 MDBMS의 구성 요소들의 기능을 정의하고 그들 간의 상호작용을 설계한다. 또한, 구성요소 중 전역 뷰 관리자 관점에서 전역 질의에 대한 결과 정보가 전역 뷰 저장소에 모두 저장되어 있는 경우, 전혀 없는 경우, 그리고 일부 저장되어 있는 경우를 비교하여 객체간의 인터페이스 및 메소드 호출에 따른 시퀀스 다이아그램을 통하여 본 시스템의 기능을 정립한다. 마지막으로, 정립된 기능을 토대로 예제 질의를 이용한 각 기능들의 수행과정을 나타내어 구체적으로 설계된 모델을 제안한다.

키워드 : 이질적인 멀티데이타베이스, 뷰, 혼용 뷰, 멀티데이타베이스 관리 시스템

A Design of Model for Interoperability in Heterogeneous Multi-Database Adopting Mixed View Management Mechanism on Distributed Environments

Lee, Seungyong[†] · Park, Jaebok[††] · Kim, Myunghee[†††] · Joo, Sujong[††††]

ABSTRACT

In this paper, we propose the MDBMS(Multi-DataBase Management System) which integrates the LDBMSs(Local DataBase Systems) with heterogeneous environment into distributed system and provides global users with rapidly query process. For designing the MDBMS, we define the functions of components and design the interaction among them. In a point of view of the global view manager in components, we describe the following 3 cases; (1)the case which the results for the global query are all stored to the global view repository, (2)the case which no result exists in the global view repository, and (3)the case which the partial results are stored to the global view repository. By comparing above cases, we establish the functionalities of our MDBMS through the sequence diagram including the interface of among objects and the method calling. Finally, we propose the model designed in the concrete by showing the executing procedures of each function using sample query on established functions mentioned above.

Key Words : Heterogeneous Multi-database, View, Mixed View, MDBMS

1. 서 론

최근들어 컴퓨터 네트워크의 급속한 발달로 인하여 각 지역 데이터베이스 시스템에서는 지역 사용자가 소유 기관의 목적에 따라 응용업무에 데이터베이스를 적용시켜 자유롭게 사용하며, 이질적이고 자치성을 가진 데이터베이스들을 논리적으로 통합하여 클라이언트에게 투명한 정보처리가 가능하도록 하고 있다[1, 2, 3, 14]. 이러한 시스템을 멀티 데이터베이스 시스템(Multi-Database System : MDBS)이라고 한다.

MDBS을 구축하였을 경우에는 지역 데이터베이스 시스템(Local Database System : LDBS)의 자치성을 보장할 수 있으므로 새로운 LDBS의 추가 및 제외가 용이한 분산 데이터베이스 시스템을 구축하는 것이 가능하다[4]. 따라서 MDBS에서는 각 LDBS이 기존의 사용자 인터페이스와 독자적인 트랜잭션 관리 기법 등을 이용하여 지역 응용 처리는 물론, 전역적 응용에 대해서는 LDBS들 간의 이질성에 대한 고려 없이 사용자가 여러 사이트에 저장된 데이터를 액세스 할 수 있도록 지원해야 한다.

※ 본 논문은 2004학년도 원광보건대학 교내연구비지원에 의하여 이루어짐.
† 종신회원 : 원광보건대학 관광경영과 교수
†† 정 회 원 : 경북과학대학 컴퓨터미디어계열(웹마스터전공) 조교수
††† 정 회 원 : 원광디지털대학교 컴퓨터정보학과 전임강사
†††† 정 회 원 : 원광대학교 컴퓨터공학과 교수
논문접수 : 2004년 11월 5일, 심사완료 : 2004년 6월 3일

이질성을 가진 각 데이터베이스를 통합하기 위해서는 이질성을 해결하기 위한 미들웨어가 필요하다. 최근 분산 이질 환경에서 여러 시스템을 통합하는데 OMG(Object Management Group)가 제안한 분산 객체 미들웨어 표준인 CORBA(Common Object Request Broker Architecture)를 주로 사용하고 있다. CORBA는 이질적인 응용시스템들의 연결을 용이하게 할 수 있는 방법을 제공할 뿐만 아니라, 분산 객체 컴퓨팅에 공통적으로 필요한 것으로 예상되는 다양한 서비스들을 제공하고 있다. 따라서 CORBA를 이용해 분산 객체 시스템을 개발할 경우 개념상으로는 구현 작업의 상당 부분이 간소화되고 손쉽게 구현할 수 있다[2, 14, 15, 16].

MDBS은 분산된 데이터베이스[7]의 투명성을 제공한다는 점에서는 분산 데이터베이스 시스템과 같지만, LDBS의 자치성 보장의 관점에서 차이를 들 수 있다. 따라서 LDB들은 서로 다른 데이터 모델과 자체 내 정보 표현 또한 다르다. 서로 다른 데이터 모델로 표현된 LDB의 정보를 그대로 사용자에게 제공할 수 없다. 서로 다른 데이터 모델로 표현된 지역 스키마(local schema)를 같은 모델을 이용하여 표현하고 공통의 데이터 모델로 표현된 지역 스키마(export schema)를 멀티 데이터베이스에서 공통으로 사용할 수 있는 전역 스키마(global schema)로 통합하는 스키마 통합과정을 거쳐야 한다.

전역 스키마를 이용하여 MDBS 사용자는 하나의 데이터베이스를 사용하듯이 전역 질의를 발생시켜 원하는 정보를 취득할 수 있다. 전역 스키마를 이용하는 전역 질의는 LDBS에 적당한 질의로의 파싱이 필요하다. 질의 처리기는 클라이언트에 의해 발생한 전역 질의를 LDB를 액세스할 수 있는 지역 질의로의 변환이 전제되어야 한다.

질의에 의해 각 LDB에 저장되어 있는 정보가 추출되면 이를 통합기에 통합시켜 클라이언트의 전역 질의에 대한 결과를 처리해야 한다.

이러한 관점에서 본 연구에서는 분산 데이터베이스의 분산성, 멀티 데이터베이스의 자치성과 통합성 그리고 웨어하우스의 데이터의 일관성을 고려하여 전역 사용자가 멀티 데이터베이스들에 저장되어 있는 데이터를 투명하고 빠른 위치 탐색과 검색을 제공할 수 있는 멀티 데이터베이스 관리 시스템(MDBMS : Multi-Database Management System)을 설계하고자 한다. 이를 위해 이전에 요청되었던 전역 질의에 대한 정보를 전역 뷰 관리자를 통해서 관리함으로써, 같은 전역 질의가 발생했을 때 LDB를 처음부터 검색하는 시간적 지연을 최소화하였다. 과거에 검색된 정보를 전역 뷰 저장소에 저장하여 놓고 이곳에서 실체 뷰[8]와 시그니쳐 뷰 인덱스[9, 10, 12] 혼용 관리 기법[5]을 적용시켜 전역 질의를 처리하기 위한 LDB 검색 비용을 감소시키고자 한다.

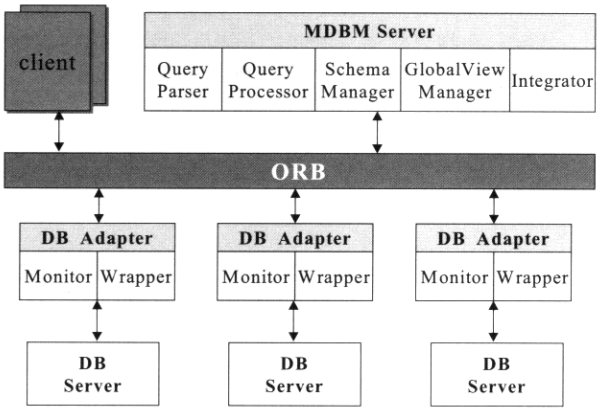
이를 위해, 본 논문의 2장에서 연구의 기반이 된 MDBMS 구조와 뷰 관리기법에 대해서 기술하고, 3장에서는 제안된 MDBMS의 기능 구조를 설계한다. 4장에서 전역 질의 처리에 대한 제안된 MDBMS의 기능적 수행과정을 예를 들어 설명하고, 5장에서 결론을 내리도록 한다.

2. 제안된 MDBMS와 뷰관리기법

이 장에서는 본 연구의 기반 구조[5]인 MDBMS의 간략한 구조와 뷰 관리를 위해 이전 연구에서 제안한 혼용 뷰 관리기법에 대해 살펴본다.

2.1 MDBMS의 구조

(그림 1)은 본 연구의 MDBMS의 구조[5]를 나타낸다. MDBMS는 클라이언트, MDBM 서버, LDB 어댑터와 LDB로 구성되며 ORB(Object Request Broker)를 통하여 통신을 수행한다.



(그림 1) 제안된 MDBMS의 구조

2.2 뷰 관리 기법

기존의 뷰 관리기법은 실체 뷰[6, 8]와 시그니쳐 뷰인덱스[7, 9, 10, 11, 12] 기법이 대표적이다. 실체 뷰 기법은 뷰를 실체화하여 저장함으로써 저장 공간을 많이 사용한다는 단점을 가지고 있는 반면에, 뷰에 대한 조건이 주어지는 경우에는 기본 릴레이션을 검색하지 않고 실체화된 내용을 이용하여 질의에 응답할 수 있으므로 신속한 응답시간을 제공한다는 장점을 가진다. 또한, 시그니쳐 뷰 인덱스 기법은 적은 저장 공간을 사용한다는 면에서 실체 뷰 기법보다 우수하지만, 기본 릴레이션의 갱신시 뷰인덱스들을 갱신하여야 하며, 인덱스에는 뷰가 사용하는 기본 릴레이션의 투플 주소만을 저장하기 때문에 뷰 조건이 주어지는 경우는 실제 릴레이션을 검색하여 비교하여야 하는 문제가 있다. 이러한 기존의 뷰 관리기법들의 장단점을 분석하여 이전 연구에서 실체 뷰와 시그니쳐 뷰인덱스의 혼용 뷰 관리기법[5]을 제안했었다.

(그림 2)는 혼용 뷰 기법을 위한 메타데이타 구조를 나타낸다.

*id*는 실체 뷰의 ID를 가리킨다. *ReplaceDataSource*는 데이터베이스나 파일 시스템등의 실체 뷰의 원래 위치를 가리키는데, 이것으로 실체 뷰가 참조한 테이블을 알 수 있고, 참조된 테이블이 변화할 경우 이를 보고하여, 실체 뷰의 유지보수를 돕는다. *QueryOp*는 이미 수행된 질의 내용을 참조한다. ViewBit는 메타데이타에서 나타내는 정보가 실체 뷰에 대한 참조(0)인지 시그니쳐 뷰인덱스에 대한 참조(1)인

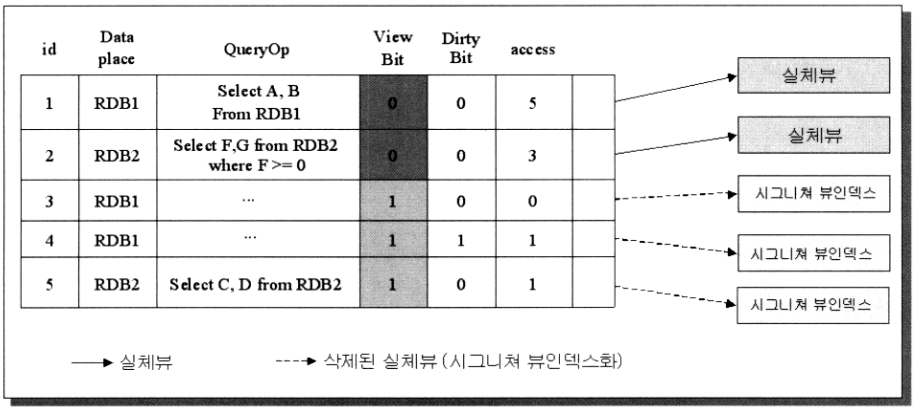
```
interface emp
{ int          id;
  int          ReplaceDataSource;
  String       QueryOp;
  int          ViewBit;
  int          DirtyBit;
  float  NoAccess;

  질의처리 method()
  질의처리 method() ...  }
```

(그림 2) 혼용 뷰 기법의 메타데이타 구조

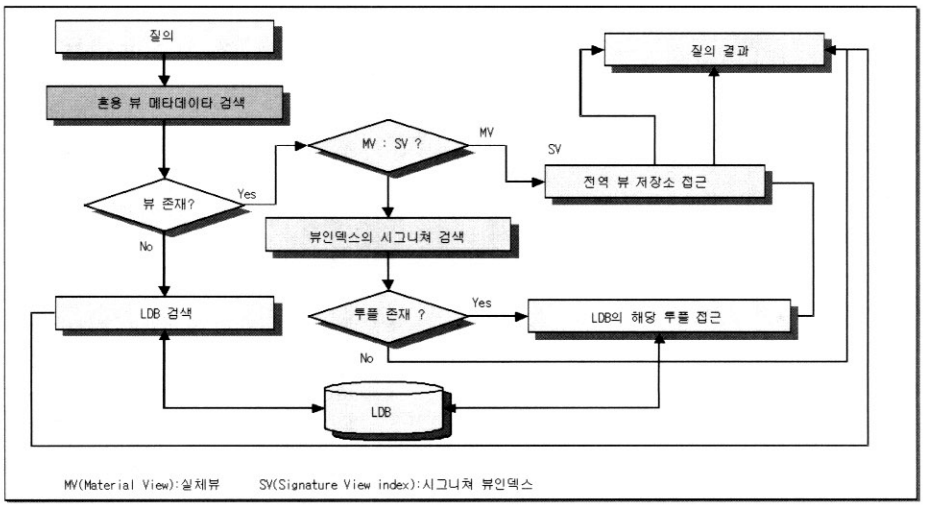
지를 구별한다. *DirtyBit*는 실체 뷰의 원위치에 있는 데이터가 변경(변형)되었는지를 나타낸다. 원위치에 있는 데이터가 변경되었을 경우에는 "1"로 표시하고, 변경되지 않았을 경우에는 "0"으로 표시한다. NoAccess는 실체 뷰와 시그니쳐 뷰인덱스에 대한 접근 횟수를 나타낸다.

(그림 3)에서 보면, 'ID 3'의 경우는 실체화되었던 뷰가 삭제되면서 시그니쳐 뷰인덱스로 정보를 보관한다. *ViewBit*는 1로 설정되어 시그니쳐 뷰인덱스임을 나타내고, *DirtyBit*는 0으로 현재 뷰에 대한 원위치의 데이터가 변경되지 않았고, *NoAccess*는 0으로 실체 뷰에서 시그니쳐 뷰인덱스화후 한번도 질의에 대한 접근이 없었음을 나타낸다. 'ID 4'의 경우는 시그니쳐 뷰인덱스에 대한 접근이 1번 있었음을 *NoAccess*로 알 수 있으며, 원위치의 데이터가 변경되었음을 *DirtyBit*값으로 나타내고 있다. 이 뷰에 대한 질의가 발생하면, 뷰는 재검색 한다. 또한 이 뷰에 대한 접근이 발생하였을 때, 검색된 뷰의 실체화 여부는 *NoAccess*값으로 판별한다. 전역 뷰 저장소에 빈 공간이 존재한다면, 뷰를 실체화시키면 된다. 그러나 전역 뷰 저장소의 빈 공간이 없다면, 현재 *NoAccess*값과 실체 뷰의 *NoAccess*값을 비교한다. 실체 뷰의 *NoAccess*값이 각각 5와 3으로 시그니쳐 뷰인덱스의 *NoAccess*값 1보다 크므로 검색된 뷰를 실체화하지 않는다. 'ID 5'의 경우는 시그니쳐 뷰인덱스에 대한 접근이 1번 있었음을 나타내며, 검색된 뷰에 대한 실체화가 이루어지지 않았음을 나타낸다.

(그림 4)는 본 모델에서 전역 뷰 관리자가 혼용 뷰 관리기법을 이용한 전역 질의에 대한 처리 수행 과정을 나타낸다.



(그림 3) 혼용 뷰 관리를 위한 메타데이터 구조의 예



(그림 4) 뷰 관리 혼용 기법

```
Mixed_View_Management Algorithm

/* input query */
 q ← query
/* check the meta-data of Mixed View */
/* view information exist in meta-data of Mixed-View */
if QueryOp_i = q in meta-data for id index i = 1,···,n then
/* test the type of view : materialized view or signature view index */
    if ViewBit_i = 0 then                    /* materialized view */
        access Global View Repository indicating id_i
        get the materialized view data
        return the result about query q
    else                                     /* signature view index */
        get the set of TID in the view index of id_i
/* check the signature information about query */
        check the signature in S(TID)_j for j = 1,···,n
/* the tuple information relating query exist */
        if S(TID)_j = the attributes of q then
            access the tuple indicating TID_j in the local DB
            return the result about query
/* the tuple information relating query not exist */
          else
            return the result 'no data'
/* view information not exist in the meta-data of Mixed-View */
    else
        access the local DBs
        check the entire local DBs for finding the result data of q
        return the result about q
```

(그림 5) 혼용 뷰 관리 알고리즘

(그림 5)는 위에서 살펴본 혼용 뷰의 메타데이타 구조를 기반으로 하는 관리 알고리즘을 나타낸다.

3. MDBMS의 구성요소 기능 정립

(그림 1)의 MDBMS를 살펴보면, 일반적인 MDBMS의 특성과 본 논문에서 제시하는 혼용뷰 관리를 효율적으로 운용할 수 있는 구성요소들을 포함하고 있다. 이번 장에서는 본 논문에서 제시하는 MDBMS의 요구사항과 이에 따라 정립된 기능들을 기술한다.

3.1 MDBMS의 요구사항

본 논문에서 MDBMS는 일반적인 멀티데이타베이스의 특성에 맞추어 이질적인 데이터베이스의 상호운용과 일관성 있는 트랜젝션 처리 및 뷰 관리자를 통한 효율적인 정보 검색 관리를 목적으로 한다. 따라서 MDBMS 모델링을 위해 기존의 다른 시스템과는 차별화된 기능별 요구사항을 다음과 같이 기술한다.

- 이질적인 데이터베이스 처리의 상호연동은 클라이언트에게 전역질의 및 질의에 대한 결과를 표현하기 위해서는 단일 인터페이스 및 GUI를 제공한다.
- 전역질의에 대한 뷰를 지역 처리하기 위해서는 전역질의에 대한 파싱 및 재구성 할 수 있는 기능 및 뷰 변경에 대한 일관성을 유지하기위한 상태를 체크할 수 있는 기능들을 제공한다(뷰 질의에 대한 파싱 및 재구성 포함).
- 전역질의에 대한 실체 뷰를 갱신할 수 있는 기능을 제공한다.
- 전역 트랜잭션에 대한 상태를 파악 처리하기 위해서는 전역 트랜잭션 모니터링을 제공한다. 또한 적절한 트랜잭션의 이루어지지 않을 경우는 철회 및 예외기능을 제공한다.
- 전역 질의를 통해 검색 및 갱신에 대한 트랜잭션도 지원한다.
- 효율적인 검색을 위한 뷰 관리에 있어서 뷰 관련 정보를 저장관리 한다.
- 질의어 처리에 있어 분산된 데이터베이스 및 스키마 관련정보를 저장 관리한다.
- 파싱된 지역 질의어에 대한 실제 LDB에 대한 질의의 전달 및 처리 결과를 보내주는 기능을 제공한다.
- LDB의 트랜잭션처리 여부를 모니터링 하기 위해 트랜잭션 상태를 트랜잭션 매니저에게 전송한다.
- 이질적인 각각의 LDB의 결과를 통합, 전역질의에 대한 최종결과를 가공할 수 있는 기능을 제공한다.
- 검색에 관련한 사항으로 뷰 질의어의 선택 및 최적화 기능을 제공한다.
- 스키마 갱신 시 뷰 매니저에 스키마의 변경을 반영해야 한다.
- 실체 뷰에 저장된 데이터에 대해 인덱스를 부여하여 데

이터 저장소에 저장되어 있는 데이터의 변경을 용이하게 관리하여 일관성을 유지해야 한다.

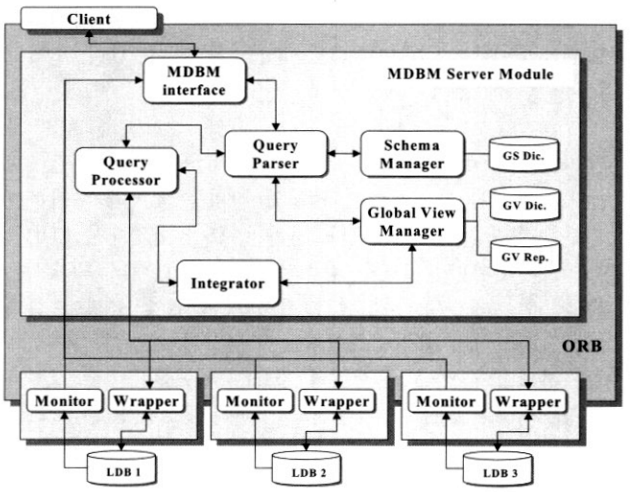
- 한 번 질의 처리된 정보는 데이터 저장소에서 처음부터 검색하지 않고 실체 뷰에 저장된 정보를 검색하여 빠른 정보 검색이 이루어지도록 해야 한다.

3.2 MDBMS의 기능정립

(그림 1)에서 본 MDBMS의 구조를 객체간 상호작용과 기능을 중심으로 세부적으로 살펴보면 (그림 6)으로 나타내어지며, ORB를 기반으로 한다.

제안된 MDBMS의 내부 구성요소를 살펴보면, 클라이언트가 MDBM 서버와 상호작용 하기 위한 인터페이스 부분으로 전역 질의를 받고, 질의에 대한 처리결과를 클라이언트에게 반환해 주는 CORBA Interface인 MDBM interface가 있다.

각각의 LDB에서 독자적으로 설계된 스키마를 통합된 형식의 전역 스키마로 제공하는 즉, 전역 스키마 관리를 위한 스키마 관리자가 있으며, 이에 대한 정보는 전역 스키마 저장소인 GS Dic.(Global Schema Dictionary)에 저장시켜 전역 데이터의 처리에 참조한다.



(그림 6) MDBMS의 기능 구조

질의 파서기(Query Parser)는 클라이언트에 의해 요구된 전역 질의를 LDB에 적합한 부분 질의로 파싱하며, 이때 스키마 관리자와 상호 작용한다.

질의 처리기(Query Processor)는 파싱된 부분 지역 질의를 각 LDB 어댑터에 전달하고 이에 대해 처리된 값을 되돌려 받아 통합기에 전달하며, 통합기에 통합된 결과들을 기반으로 클라이언트가 요구한 질의를 처리한다.

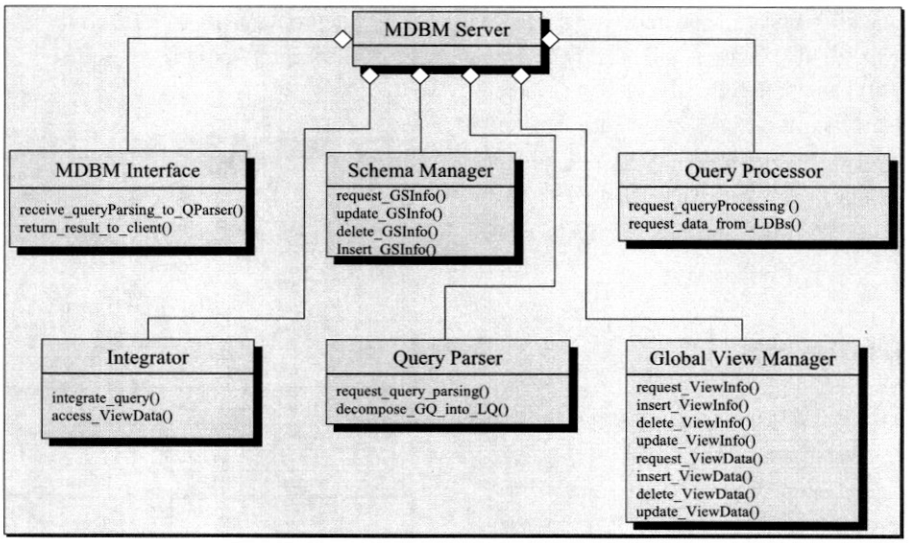
멀티 데이터베이스에서의 빠른 검색을 위한 본 연구의 핵심부분인 전역 뷰 관리자(Global View Manager)는 전역 질의에 의해 처리된 이력이 있는 전역 질의에 대한 정보를 GV Dic.에 저장시키고, LDB에서 검색된 결과들은 GV Rep.(Global View Repository)에 저장하여 전역적으로 관리한다.

LDB 어댑터(Adapter)에 있는 랩퍼(Wrapper)는 각 LDB에 적합한 형태로 부여된 부분 질의를 LDBMS에 전달하여 처리된 결과를 질의 처리기에 반환한다. 모니터(Monitor)는 LDB와 GV Rep.에 저장되어 있는 데이터들의 일관성을 유지하기 위해 LDB를 모니터링하여 변경사항을 GV Rep.에 반영하도록 한다. 본 논문에서는 일관성 유지를 위한 객체들의 상호작용은 뷰관리 기법을 제시하고자 하는 연구 취지에 맞춰 생략한다.

(그림 7)은 MDBM 서버의 모든 구성 객체들에 대한 클래스 다이어그램을 나타낸다.

본 논문에서는 MDBMS의 구성 요소들 중 전역 뷰 관리자에 대한 기능 명세와 메소드 정의를 기술한다.

전역 뷰 관리자는 GV Dic.와 GV Rep.를 관리하며, 과거에 질의되었던 뷰들에 대한 관리기법으로 실체 뷰와 시그니처 뷰인덱스 기법을 혼용함으로써 분산 환경의 멀티 데이터



(그림 7) MDBM 서버의 클래스 다이어그램

베이스에 대한 질의 성능을 향상시킨다. 전역 뷰 관리자는 질의 파서기로부터 전역 질의 및 부분 질의를 전달받아 과거에 발생되었던 질의인지를 GV Dic.에 저장되어 있는 혼용 뷰 메타데이터의 정보를 이용하여 판단한다. GV Dic.에 저장된 혼용 뷰 메타데이터의 정보를 이용하여 질의들에 대한 뷰의 종류를 판별하여 그에 맞는 데이터 검색 방법을 판단한다.

만일 실체 뷰로 존재하면 GV Rep.로 접근을 시도하고, 시그니처 뷰인덱스로 존재하면 먼저 시그니처 뷰인덱스 메타데이터 정보를 이용하여 원하는 결과에 대한 튜플과 해당 조건을 검색한 후, LDB에 접근한다. 여기에서 전역 뷰 관리자가 LDB에 접근하는 세부적인 과정은 생략한다.

전역 뷰 관리자는 뷰들의 존재유형의 변화에 따른 관리를 위해 GV Dic.에 있는 혼용 뷰 메타데이터를 변경한다. 전역 뷰 관리자는 LDB로부터 새로 검색되어진 뷰에 대한 존재 유형을 판단하여 관리한다. 이때 새로 검색되어진 뷰들에 대한 정보는 통합기로부터 전달받고 이 뷰에 대한 정보는 혼용 뷰 메타데이터에 삽입된다.

전역 뷰 관리자는 뷰의 존재 유형이 실체 뷰로 결정된 뷰에 대해서는 GV Rep.에 저장하여 관리한다. 물론 이때 이에 대한 정보는 혼용 뷰 메타데이터에 삽입된다.

GV Dic.에 저장된 혼용 뷰 메타데이터 구조는 앞에서 정의한 형태를 사용한다. 따라서, 전역 뷰 관리자는 이 구조를 이용하여 GV Rep.의 실체 뷰들을 유지관리하며 저장된 실체 뷰들의 접근 회수 등을 기반으로 관리한다.

전역 뷰 관리자의 기능 명세에 따른 메소드 정의는 (그림 7)의 클래스 다이어그램에서 Global View Manager 부분으로 나타난다. Global View Manger는 전역 뷰들에 대한 메타 정보를 관리하기 위해, GV Dic.에 저장되어 있는 전역 뷰에 대한 정보를 검색하기 위한 request_ViewInfo(), 새로운 전역 뷰에 대한 정보를 저장하기 위한 insert_ViewInfo(), GV Dic.에서 더 이상 뷰로 존재하지 않는 전역 뷰에 대한 정보를 삭제하기 위한 delete_ViewInfo(), 기존의 전역 뷰에 대한 정보를 수정하기 위한 update_ViewInfo()를 정의한다. 또한, GV Rep.에 저장되어 있는 실체 뷰들을 관리하기 위하여, 실체 뷰의 데이터를 액세스할 수 있는 request_ViewData(), 새로운 실체 뷰를 저장하기 위한 insert_ViewData(), 더 이상 실체 뷰로 존재하지 않는 뷰를 삭제하기 위한 delete_ViewData(), 저장되어 있는 실체 뷰의 일부 데이터 변경에 따른 수정을 위한 update_ViewData()등의 메소드를 정의한다.

4. 전역 질의 관리 기능 정립

이 장에서는 앞에서 정의된 MDBMS의 구성요소들의 기능을 토대로 이질적인 멀티 데이터베이스간의 연동을 효율적으로 지원하고, 혼용 뷰 관리 기법을 통하여 전역 질의에 대한 메타 데이터 및 뷰를 관리하여 클라이언트가 요구하는 전역 질의에 대한 결과를 얻기 위한 빠른 검색을 제공하는 MDBMS의 기능적 수행과정을 설명한다.

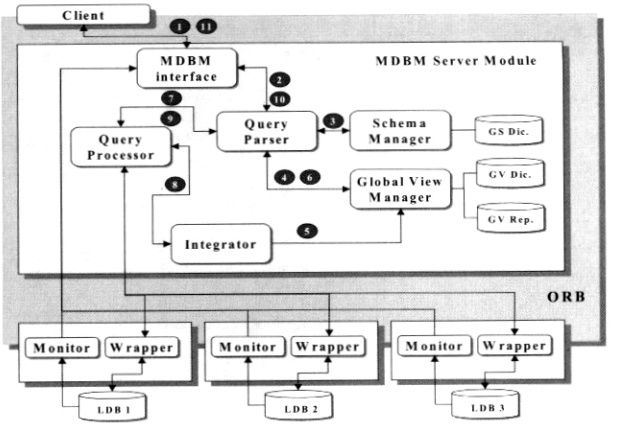
4.1 전역 질의 처리 절차

MDBMS에서 클라이언트의 전역 질의를 처리하는 절차는 3가지 방법으로 분류한다. 클라이언트의 전역 질의를 질의 파서기가 부분 질의로 파싱한 결과를 토대로 GV Rep.에 질의 조건 데이터가 어느 정도 존재하는지의 여부를 기반으로 분류한다. 첫째 방법은 클라이언트의 전역 질의의 파싱된 부분 질의들에 대한 요구 데이터가 모두 이전에 질의되었던 것으로 뷰의 메타데이터에 정보를 가지고 있는 경우이다. 이때 MDBMS는 질의에 대한 해당 데이터를 얻기 위해 LDB 전체를 검색할 필요가 없다. 둘째 방법은 파싱된 부분 질의들에 대한 요구 데이터가 일부 뷰로서 존재하는 경우이다. 이때 뷰로 존재하지 않는 부분 질의들에 대한 처리는 LDB를 검색하여야 한다. 세 번째 방법은 파싱된 부분 질의들에 대한 요구 데이터가 모두 뷰로서 존재하지 않는 경우이다. 이 경우는 클라이언트들이 이와 관련된 질의를 과거에 한 적이 없음을 의미한다. 이때는 LDB 전체를 검색하여야 하므로, 본 시스템에서 제시하는 혼용 뷰 관리 모델의 장점이 부각되지 않는다.

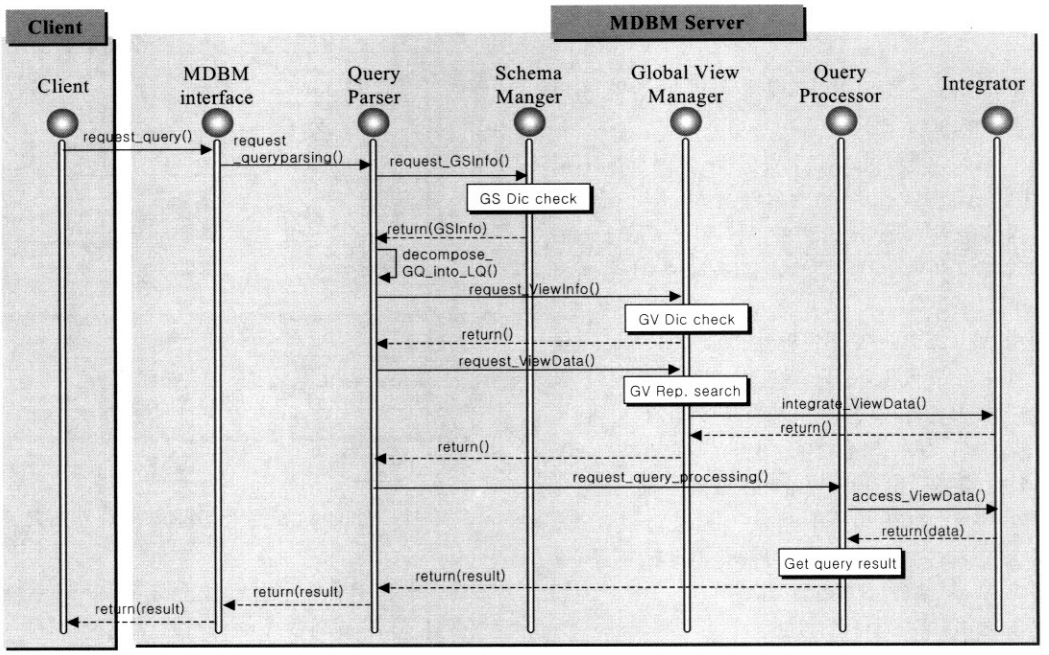
4.1.1 모든 데이터가 GV Rep.에 저장되어 있는 경우

이 경우에는 클라이언트의 전역 질의를 처리하기 위하여 LDB에 접근 할 필요가 없다. 따라서, 분산 환경에서 LDB에 접근하기 위한 입/출력을 감소시킴으로써 데이터 액세스를 빠르게 할 수 있다. 여기에서 전역 뷰의 존재 유형이 실체 뷰인지 시그니처 뷰인덱스인지는 구별하지 않겠다. 본 절에서는 전체적인 MDBM 서버의 기능 수행에 따른 각 객체들의 상호작용에 초점을 맞추어 설명한다. (그림 8)은 클라이언트의 전역 질의 처리에 필요한 모든 뷰가 전역 뷰로 존재하는 경우의 처리 과정이다.

① 클라이언트가 MDBM interface에게 전역 질의를 보낸다.
② MDBM interface는 클라이언트에게서 받은 전역 질의를 질의 파서기에게 전달한다.



(그림 8) ALL : 모든 부분 질의에 대한 뷰가 존재



(그림 9) ALL인 경우의 시퀀스 다이어그램

③ 질의 파서기는 클라이언트의 전역 질의를 처리하기 위해, 스키마 관리자에게 전역 질의의 부분 질의 분해에 필요한 정보를 얻어온다. 스키마 관리자는 GS Dic.에 있는 전역 스키마에 대한 정보를 이용하여 질의 파서기의 요구를 처리한다.

④ 질의 파서기는 스키마 관리자와의 상호작용으로 전역 질의를 처리할 수 있는 부분 질의로 변환한 후, 이 질의들에 대한 기존의 재질의 여부를 판별하기 위해 전역 뷰 관리자와 상호 작용한다. 전역 뷰 관리자는 GV Dic.에 있는 혼용 뷰 메타데이터를 이용하여 질의에 대한 뷰가 존재하는지 여부를 판별한다.

⑤ 만일 질의에 대한 뷰가 이미 존재한다면, 전역 뷰 관리자는 GV Rep.에서 해당되는 뷰들을 통합기로 통합시킨다.

⑥ 전역 뷰 관리자는 질의 파서기에게 요청한 질의에 대한 모든 뷰를 통합기로 통합시켰음을 알려준다.

⑦ 질의 파서기는 질의 처리기에게 전역 질의에 대한 결과 데이터를 얻기위해, 통합기에 통합된 뷰에 대한 질의 처리를 요청한다.

⑧ 질의 처리기는 통합기에 통합된 뷰들의 연산을 통하여 원하는 결과 데이터를 얻어낸다.

⑨ 질의 처리기는 ⑧에서 얻어진 질의 결과를 질의 파서기에게 전달한다.

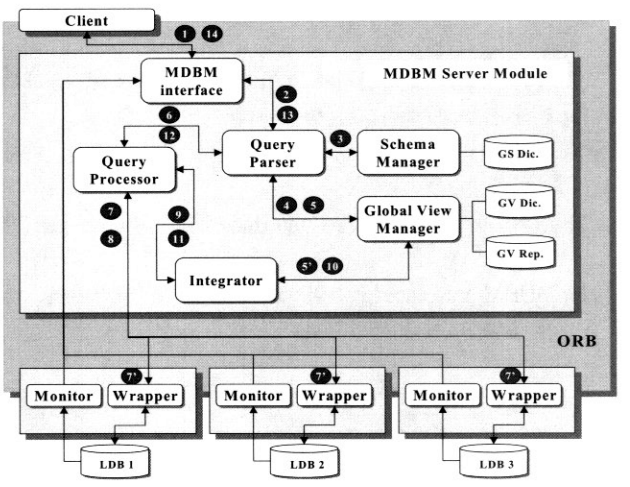
⑩ 질의 파서기는 MDBM interface에게 질의 결과를 반환한다.

⑪ MDBM interface는 전역 질의를 요청한 클라이언트에게 결과 데이터를 반환한다.

(그림 9)는 모든 부분 질의에 대한 뷰가 전역 뷰 저장소에 존재하는 경우 각 객체들 간의 상호 인터페이스를 시퀀스 다이어그램으로 표현한 것이다. (그림 9)를 살펴보면, 클라이언트의 전역 질의를 처리하기 위해, LDBS과의 상호작용이 수행되지 않는다. 따라서 클라이언트에게 빠른 결과를 반환시킬 수 있다.

4.1.2 일부 데이터가 GV Rep.에 저장되어 있는 경우

이 경우에는 클라이언트의 전역 질의에 대한 일부 부분 질의를 처리하기 위하여 LDB에 접근해야 한다. 그러나 전역 뷰로 존재하는 데이터에 대해서는 LDB에 접근 할 필요가 없으므로, 일반적인 MDBS보다는 입/출력 횟수가 줄어든다. (그림 10)은 클라이언트의 전역 질의 처리에 필요한 일부 뷰가 전역 뷰로 존재하는 경우의 처리 과정이다.

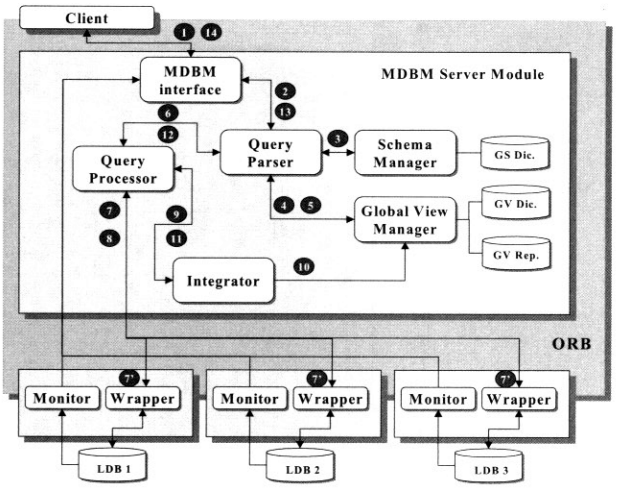


(그림 10) Partial : 일부 부분 질의에 대한 뷰만 존재

① 클라이언트가 MDBM interface에게 전역 질의를 보낸다.
② MDBM interface는 클라이언트에게서 받은 전역 질의를 질의 파서기에게 전달한다.
③ 질의 파서기는 클라이언트의 전역 질의를 처리하기 위해, 스키마 관리자에게 전역 질의의 부분 질의 분해에 필요한 정보를 얻어온다. 스키마 관리자는 GS Dic.에 있는 전역 스키마에 대한 정보를 이용하여 질의 파서기의 요구를 처리한다.
④ 질의 파서기는 스키마 관리자와의 상호작용으로 전역 질의를 처리할 수 있는 부분 질의로 변환한 후, 이 질의들에 대한 기존의 재질의 여부를 판별하기 위해 전역 뷰 관리자와 상호 작용한다. 전역 뷰 관리자는 GV Dic.에 있는 혼용 뷰 메타데이터를 이용하여 질의에 대한 뷰가 존재하는지 여부를 판별한다.
⑤ 전역 뷰 관리자는 질의 파서기로부터 전달된 부분 질의들 중의 일부 부분 질의들에 대해서만 뷰가 존재한다는 것을 질의 파서기에게 전달한다.
⑤' ⑤와 동시에 전역 뷰 관리자는 GV Rep.에 존재하는 부분 질의에 해당되는 뷰들을 통합기로 보낸다.
⑥ 질의 파서기는 질의 처리기에게 GV Rep.에 존재하지 않는 뷰에 대한 부분 질의들에 대해서, LDB로부터 직접 검색해야 함을 요청한다.
⑦ 질의 처리기는 부분 질의에 해당되는 LDB 어댑터에 있는 랩퍼에게 부분 질의들에 대한 결과를 얻기 위한 데이터를 요청한다.
⑧ 각 랩퍼는 LDB를 검색하여 질의 처리기가 요청한 질의에 대한 뷰를 반환한다.
⑨ 질의 처리기는 각 LDB 어댑터의 랩퍼로부터 반환받은 뷰를 통합기에 전달하여, GV Rep.로부터 전달되어 온 뷰들과 통합한다. 이때 전역 질의를 부분 질의로 분해할 때 나온 정보들을 이용하여 부분 질의들에 대한 뷰들을 통합한다.
⑩ 통합기는 LDB에서 얻어진 뷰들의 정보를 전역 뷰 관리자에게 보냄으로써, 실체 뷰/뷰인덱스 혼용 기법에 의해 관리한다.
⑪ 질의 처리기는 통합기에 통합된 뷰들의 연산을 통하여 원하는 결과 데이터를 얻어낸다.
⑫ 질의 처리기는 ⑪에서 얻어진 질의 결과를 질의 파서기에게 전달한다.
⑬ 질의 파서기는 MDBM interface에게 질의 결과를 반환한다.
⑭ MDBM interface는 전역 질의를 요청한 클라이언트에게 결과 데이터를 반환한다.

4.1.3 모든 데이터가 GV Rep.에 저장되어 있지 않은 경우

이 경우에는 클라이언트의 전역 질의를 처리하기 위하여 해당되는 LDB에 모두 접근해야 한다. 따라서 분산 환경에서 MDBMS가 주는 장점이 부각되지 않는다. 그러나 이 과정을 거쳐 검색되어진 뷰들은 전역 뷰로 존재하게 되므로 이후의 관련 질의들에 대해서는 빠른 검색을 제공할 수 있다. (그림 11)은 클라이언트의 전역 질의 처리에 필요한 모든 뷰가 전역 뷰로 존재하지 않는 경우의 처리 과정이다.



(그림 11) Nothing : 모든 부분 질의에 대한 뷰가 존재하지 않음

① 클라이언트가 MDBM interface에게 전역 질의를 보낸다.
② MDBM interface는 클라이언트에게서 받은 전역 질의를 질의 파서기에게 전달한다.
③ 질의 파서기는 클라이언트의 전역 질의를 처리하기 위해, 스키마 관리자에게 전역 질의의 부분 질의 분해에 필요한 정보를 얻어온다. 스키마 관리자는 GS Dic.에 있는 전역 스키마에 대한 정보를 이용하여 질의 파서기의 요구를 처리한다.
④ 질의 파서기는 스키마 관리자와의 상호작용으로 전역 질의를 처리할 수 있는 부분 질의로 변환한 후, 이 질의들에 대한 기존의 재질의 여부를 판별하기 위해 전역 뷰 관리자와 상호 작용한다. 전역 뷰 관리자는 GV Dic.에 있는 혼용 뷰 메타데이터를 이용하여 부분 질의들에 대한 뷰가 존재하는지 여부를 판별한다.
⑤ 전역 뷰 관리자는 질의 파서기로부터 전달된 부분 질의들 모두가 이전에 질의된 적이 없음을 질의 파서기에게 전달한다. 이것은 부분 질의들에 대한 뷰가 GV Rep.에 존재하지 않는다는 것을 의미한다.
⑥ 질의 파서기는 질의 처리기에게 전역 질의에 대한 모든 부분 질의들을 처리하기 위해, LDB로부터 직접 검색해야 함을 요청한다.
⑦ 질의 처리기는 부분 질의를 처리할 LDB 어댑터에 있는 랩퍼에게 부분 질의들에 대한 결과를 얻기 위한 데이터를 요청한다.
⑧ 각 랩퍼는 LDB를 검색하여 질의 처리기가 요청한 부분 질의에 대한 뷰를 반환한다.
⑨ 질의 처리기는 각 LDB 어댑터에 있는 랩퍼로부터 반

환받은 뷰를 통합기에 통합한다. 이때 전역 질의를 부분 질의로 분해할 때 나온 정보들을 이용하여 부분 질의들에 대한 뷰를 통합한다.

⑩ 통합기는 LDB에서 얻어진 뷰들의 정보를 전역 뷰 관리자에게 보냄으로써, 이들을 실체 뷰/뷰인덱스 혼용 기법에 의해 관리한다.

⑪ 질의 처리기는 통합기에 통합된 뷰들의 연산을 통하여 원하는 결과 데이터를 얻어낸다.

⑫ 질의 처리기는 ⑪에서 얻어진 질의 결과를 질의 파서기에게 전달한다.

⑬ 질의 파서기는 MDBM interface에게 질의 결과를 반환한다.

⑭ MDBM interface는 전역 질의를 요청한 클라이언트에게 결과 데이터를 반환한다.

(그림 12)는 모든 부분 질의에 대한 뷰가 전역 뷰 저장소에 존재하지 않는 경우 각 객체들간의 상호 인터페이스를 시퀀스 다이어그램으로 표현한 것이다. (그림 12)를 살펴보면, 클라이언트의 전역 질의를 처리하기 위해, LDBS과의 상호 작용이 수행된다. 이 경우에는 전역 뷰 관리기법에 따른 이점을 얻을 수 없는 최악의 경우에 해당된다.

4.2. 전역질의 처리과정에 대한 예제

이 절에서는 본 연구에서 제시한 MDBMS의 구조를 바탕으로 질의를 처리하는 과정을 도서대출 데이터베이스 예제를 통하여 설명한다.

먼저, 예제로 살펴볼 스키마는 다음 <표 1>과 같다.

<표 1>를 보면, DB1은 클라이언트가 책을 대출한 기록에 대한 데이터베이스이다. LC_NO는 책 번호이며, CARD_NO는 클라이언트의 카드 번호를 나타내고, DATE는 클라

〈표 1〉 예제 스키마

DB1	
	LOANS(LC_NO, CARD_NO, DATE)
DB2	
	BORROWERS(CARD_NO, NAME, ADDR, CITY)
DB3	
	BOOKS(LC_NO, TITLE, AUTHOR, P_NAME)

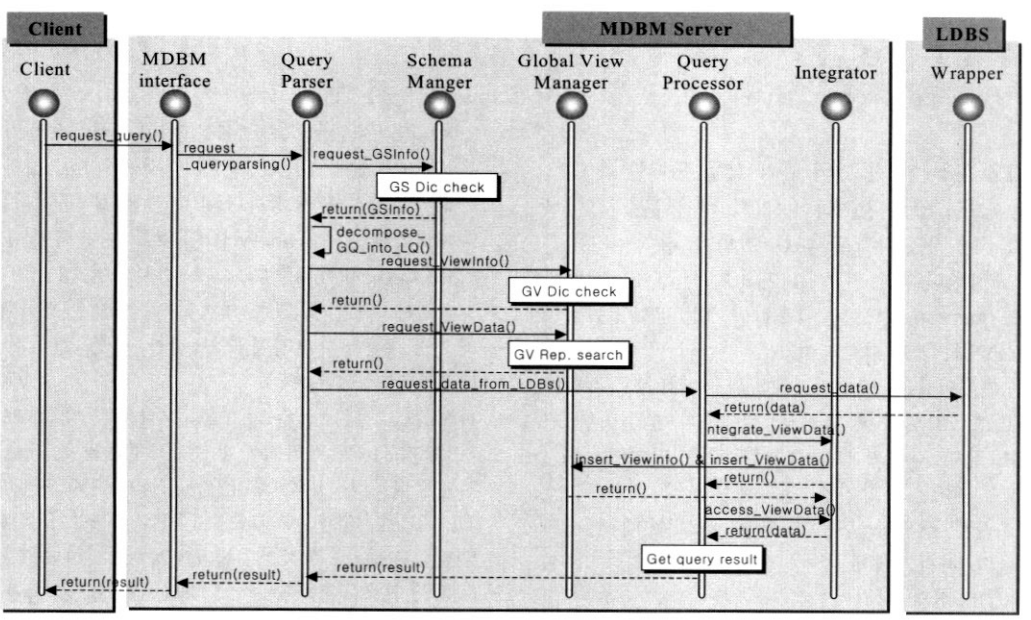
이언트가 그 책을 대출한 날짜이다. DB2는 클라이언트들에 대한 데이터베이스이다. CARD_NO는 클라이언트의 카드 번호, NAME은 클라이언트의 이름이고, ADDR은 주소, CITY는 도시이다. DB3은 책에 대한 데이터베이스이다. LC_NO는 책의 고유번호이고, TITLE은 책의 제목이며, AUTHOR은 저자, P_NAME은 출판사명이다. 이렇게 하나의 도서대출 데이터베이스의 릴레이션들이 3개의 데이터베이스로 구성되어 있는 환경에서 MDBMS의 내부 구성 요소들이 어떻게 작동하는지 설명한다.

위의 스키마를 바탕으로 클라이언트가 97년 1월 1일 이전에 빌려간 책들에 대한 목록을 얻기 위해 다음 (그림 13)과 같은 질의를 보낸다고 가정한다.

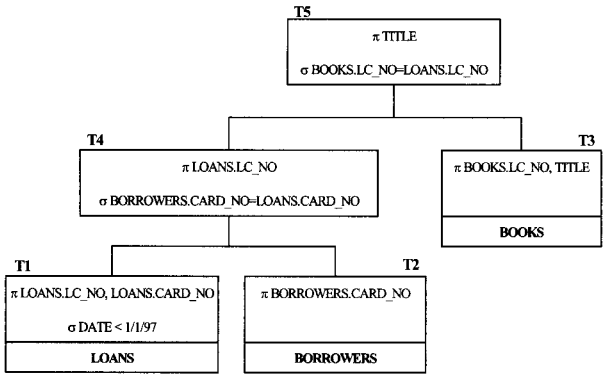
클라이언트가 보낸 질의는 MDBM interface에게 전달된다. MDBM interface는 클라이언트의 질의를 질의 파서기에

```
SELECT BOOKS.TITLE
FROM BOOKS
WHERE BOOKS.LC_NO =
      (SELECT LOANS.LC_NO
       FROM LOANS, BORROWERS
       WHERE LOANS.DATE <1/1/97
       AND BORROWERS.CARD_NO = LOANS.CARD_NO)
```

(그림 13) 예제에 대한 질의



(그림 12) Nothing인 경우의 시퀀스 다이어그램



(그림 14) 전역 질의를 부분 질의로 파싱한 구조

게 전달한다. 질의 파서기는 클라이언트가 요청한 전역 질의에 대한 파싱을 수행하여 부분 질의로 분해한다. (그림 13)에 대한 전역 질의를 부분 질의로 파싱한 구조는 다음 (그림 14)와 같다. 여기에서 5개의 부분으로 질의를 분해하였으며, T1은 LOANS 릴레이션에서 97년 1월 1일 이전에 대출한 대출자의 카드번호와 책 번호를 얻기 위한 질의이다. T2는 BORROWERS 릴레이션에서 대출자의 카드 번호를 얻기 위한 질의이다. T3는 BOOKS 릴레이션에서 책의 고유번호와 책의 제목을 얻기 위한 질의이다. T1, T2, T3는 각각 실질적인 데이터에 대한 질의 요청에 해당된다. T4와 T5는 T1, T2, T3들의 데이터와 처리되는 부분들로 T4는 T1과 T2의 처리결과를 이용하여 대출된 책의 고유번호를 얻어낸다. 여기에서 얻어진 결과와 T3에 의해 얻어진 결과를 이용하여 T5를 수행하는데 그 결과로 97년 1월 1일 이전에 대출된 책의 목록을 얻는다.

여기에서 본 연구의 뷰 관리 기법의 해당 경우에 따라 크게 세 가지 종류의 방법으로 질의에 대한 처리 과정을 수행할 수 있다. 이들에 대한 종류별 처리 과정을 4.1절의 기능정립을 참고하여 설명한다. 이때 4.1절 각각의 기능정립에서 ①부터 ③까지는 모두 같은 과정을 수행하며, ③의 과정을 수행한 결과는 (그림 14)와 같이 나타난다.

4.2.1 T1, T2, T3에 대한 전역 뷰가 모두 존재할 때

이 경우에는 (그림 8)을 참조하여, 질의 파서기는 전역 뷰 관리자에게 T1, T2, T3 각각에 대한 질의가 전역 뷰로 관리되었는지를 요청한다(④). 전역 뷰 관리자는 혼용 뷰 메타데이터에 있는 'queryop' 정보를 이용하여 T1, T2, T3 질의에 대한 뷰가 존재하는지 검사하여, 부분 질의들 모두에 대한 뷰가 이미 존재함을 판별한다. 해당되는 각각의 질의에 대한(즉, T1, T2, T3) 뷰를 통합기로 전달한다(⑤). 또한 전역 뷰 관리자는 질의 파서기에게 요청한 부분 질의(T1, T2, T3)에 대한 모든 뷰를 통합기에 전달했음을 통보한다(⑥). 질의 파서기는 전역 질의를 부분 질의로 분해할 때 발생된 정보와 함께 질의 처리기에게 부분 질의들에 대한 처리 요청을 전달한다(⑦). 질의 처리기는 통합기에 통합된 부분 질의들에 대한 질의 처리 T4와 T5를 수행하여(⑧) 처리된 결과를 질의 파서기에게 전달한다(⑨). 질의 파서기는 MDBM interface에게 클라이언트가 요청한 전역 질의에 대한 결과를 반환하고(⑩), MDBM interface는 클라이언트에게 이를 반환한다(⑪).

4.2.2 T1, T2에 대한 전역 뷰만 존재할 때

이 경우에는 (그림 10)을 참조하여, 질의 파서기는 전역 뷰 관리자에게 T1, T2, T3 각각에 대한 질의가 전역 뷰로 관리되었는지를 요청한다(④). 전역 뷰 관리자는 혼용 뷰 메타데이터에 있는 'queryop' 정보를 이용하여 T1, T2, T3 질의에 대한 뷰가 존재하는지 검사하여, T1과 T2의 부분 질의들에 대한 뷰가 이미 존재하고 T3는 질의된 적이 없음을 판별한다. 전역 뷰 관리자는 해당되는 각각의 질의에 대한(즉, T1, T2) 뷰를 통합기로 전달한다(⑤'). 또한 전역 뷰 관리자는 질의 파서기에게 요청한 부분 질의(T1, T2, T3)에 대해 T3는 전역 뷰로 존재하지 않으며, T1과 T2에 대한 뷰는 통합기에 전달했음을 통보한다(⑤). 질의 파서기는 전역 질의를 부분 질의로 분해할 때 발생된 정보와 함께 질의 처리기에게 부분 질의들에 대한 처리 요청을 전달한다(⑥). 질의 처리기는 먼저 전역 뷰로 존재하지 않는 T3에 대한 데이터를 LDB에서 얻기 위해 지역 DB1의 랩퍼에게 요청한다(⑦). 지역 DB1의 랩퍼는 LDBMS와 상호 작용하여 T3에 대한 데이터를 질의 처리기에게 반환한다(⑧). 질의 처리기는 ⑧에서 반환받은 데이터를 통합기에 저장하고(⑨), 통합기는 전역 뷰 관리자에게 새로운 데이터가 검색되었음을 통보한다(⑩). 전역 뷰 관리자는 관리기법에 따라 새로 검색된 데이터를 관리한다. 질의 처리기는 통합기에 저장되어 있는 부분 질의들에 대한 질의 처리 T4와 T5를 수행하여(⑪) 처리된 결과를 질의 파서기에게 전달한다(⑫). 질의 파서기는 MDBM interface에게 클라이언트가 요청한 전역 질의에 대한 결과를 반환하고(⑬), MDBM interface는 클라이언트에게 이를 반환한다(⑭).

4.2.3 T1, T2, T3에 대한 전역 뷰가 모두 존재하지 않을 때

이 경우에는 (그림 11)을 참조하여, 질의 파서기는 전역 뷰 관리자에게 T1, T2, T3 각각에 대한 질의가 전역 뷰로 관리되었는지를 요청한다(④). 전역 뷰 관리자는 혼용 뷰 메타데이터에 있는 'queryop' 정보를 이용하여 T1, T2, T3 질의에 대한 뷰가 존재하는지 검사하여, T1, T2, T3의 부분 질의들에 대한 뷰가 존재하지 않음을 판별한다. 전역 뷰 관리자는 질의 파서기에게 요청한 부분 질의(T1, T2, T3)에 대해 모두 전역 뷰로 존재하지 않음을 통보한다(⑤). 질의 파서기는 전역 질의를 부분 질의로 분해할 때 발생된 정보와 함께 질의 처리기에게 부분 질의들에 대한 처리 요청을 전달한다(⑥). 질의 처리기는 전역 뷰로 존재하지 않는 T1, T2, T3에 대한 데이터를 LDB에서 얻기 위해 지역 DB1, DB2, DB3의 각각의 랩퍼에게 요청한다(⑦). 지역 DB1, DB2, DB3의 랩퍼는 각 LDBMS와 상호 작용하여 질의 처리 데이터를 질의 처리기에게 반환한다(⑧). 질의 처리기는

⑧에서 반환받은 데이터들을 모두 통합기에 저장하고(⑨), 통합기는 전역 뷰 관리자에게 새로운 데이터들이 검색되었음을 통보한다(⑩). 전역 뷰 관리자는 관리기법에 따라 새로 검색된 데이터들을 관리한다. 질의 처리기는 통합기에 저장되어 있는 부분 질의들에 대한 질의 처리 T4와 T5를 수행하여(⑪) 처리된 결과를 질의 파서기에게 전달한다(⑫). 질의 파서기는 MDBM interface에게 클라이언트가 요청한 전역 질의에 대한 결과를 반환하고(⑬), MDBM interface는 클라이언트에게 이를 반환한다(⑭).

5. 결 론

본 연구는 뷰에 대한 질의 처리 시 디스크 입/출력을 감소와 LDB에 접근하는 횟수를 줄이고, 분할된 부분 질의들의 연산 결과를 실체뷰와 시그니쳐 뷰인덱스 혼용 기법으로 관리함으로써 질의 수행 횟수를 감소시킬 수 있다는 기존의 연구를 토대로 혼용뷰 관리기법을 적용시켜 검증할 수 있는 모델을 설계하고자 하였다.

이에 따라, 분산환경에서 실체뷰와 시그니쳐 뷰인덱스 기법을 적용시키는 MDBMS를 모델링하여 이에 대한 알고리즘을 작성하고 뷰에 전역 질의에 대한 데이터의 보유 여부에 따라 각 시스템의 구성 요소들을 기능정립하고 상황별에 따른 처리 절차를 제시하였다. 또한, 예제를 통해 모델링된 MDBMS의 구성요소 기능 및 기능 수행을 위한 각 구성요소간의 제시된 절차를 설명하였다.

앞으로의 연구방향은 설계된 MDBMS를 구현하여 연구된 뷰 관리 기법을 적용하여 일반적인 관리 기법과의 우수성을 검증하고자 한다. 또한, 본 연구에서는 논외로 거론되었던 LDB에서 데이터 변경을 뷰에 반영하여 LDB와 뷰와의 데이터 일관성을 유지시키기 위한 방안들에 대한 연구를 계속하고자 한다.

참 고 문 헌

[1] A. Dogac, C. Dengi, E. Kilic, G. Ozhan, F. Ozcan, S. Nural, C. Evrendilek, U. Halici, B. Arpinar, P. Koksal, N. Kesim, S. Mancuhan, "METU Interoperable Database System", ACM SIGMOD Record, Vol.24, No.3, September, 1995.

[2] E. Kilic, G. Ozhan, C. Dengi, N. Kesim, P. Koksal, A. Dogac, "Experience in Using CORBA for a Multidatabase Implementation", Proc. Of 6th International Workshop on Database and Expert System Applications, London Sept., 1995.

[3] W. Litwin and A. Abdellatif, "Multidatabase Interoperability", 1986.

[4] M. Tamer Zsu, P. Valduries, Principles of Distributed Database Systems, Prentice-Hall International Inc., 1991.

[5] 이승용, 김명희, 주수종, "CORBA 기반에서 효율적인 질의 처리를 위한 실체뷰와 시그니쳐 뷰인덱스의 혼용", 한국정보처리학회지 Vol.11, No.1, 2004.

[6] S. Chaudhuri, R. Krishnamurthy, S. Potaminanos, and K. Shim, "Materialized Views Optimizing Queries with Materialized Views", IEEE, 1995.

[7] 용환승, "관계 데이터베이스에서 시그니쳐를 이용한 뷰인덱스 기법", 정보처리논문지 제3권 제4호, 1996.

[8] A. Kawaguchi, D. Lieuwen, I. Mumick, D. Quass, K. Ross, "Concurrency Control Theory for Deferred Materialized Views", In Proc. of the International Conference on Database Theory, Athens, Greece, January 1997.

[9] F. Rabitti and P. Savino, "Image Query Processing Based on Multi-level Signatures", in Proc. of ACM SIGIR Conf. on Research and Development in Information Retrieval, pp.305-314, October, 1991.

[10] H. S. Yong, S. Lee, and H. J. Kim, "Applying Signatures for Forward Traversal Query Processing in Object-Oriented Databases", in International Conference on Data Engineering, pp.518-525, Feb., 1994.

[11] N. Roussopoulos, "An Incremental Access Method for ViewCache: Concept, Algorithms and Cost Analysis", ACM Trans. on Database Systems, Vol.16, No.3, pp.535-563, 1991.

[12] P. Zezula et al. "Dynamic Partitioning of Signature Files", ACM Trans. on Information Systems, Vol.9, No.4, pp.336-369, 1991.

[13] D. Lee and C. Leng, "Partitioned Signature File : Design issues and performance evaluation", ACM Trans. on Office Information Systems, Vol.7, pp.212-223, April, 1989.

[14] 김지운, 김보경, 이근철, 문강식, 이전영, "CORBA 기반 멀티 데이터베이스 관리 시스템의 설계", Available from http://iislab.postech.ac.kr/mrg/publish/xwidd/kiss.doc.

[15] 문창주, 이선정, 박성공, 백두권, "CORBA 환경에서 메타데이터를 이용한 관계형 데이터베이스와 파일 시스템간의 상호운용성 해결 방안", 한국정보과학회 '98 봄 학술발표 논문집(B), 25권 1호, 1998.

[16] 변광준, "CORBA 기반의 이질적인 데이터베이스 통합", 한국정보과학회지 제17권 제7호, 1999.

이 승 용

e-mail : dragon@wkhc.ac.kr

1986년 원광대학교 전자공학과(공학사)

1990년 조선대학교 대학원 전산기공학과 (공학석사)

2002년 원광대학교 대학원 컴퓨터공학과 (공학박사)

1991년~2004년 원광보건대학 컴퓨터응용개발과 교수

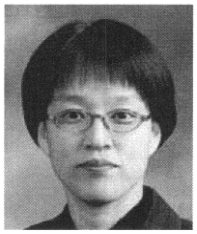
2004년~현재 원광보건대학 관광경영과 교수

관심분야 : 멀티 데이터베이스, 분산시스템, 데이터통신, 인터넷 통신

박 재 복

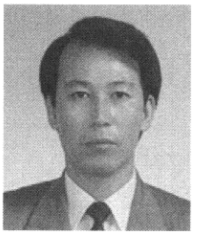
e-mail : jbpark@kbsc.ac.kr
1969년 계명대학교 가정관리학과(학사)
1989년 경북산업대학교 전산학과(공학사)
1989년 계명대학교 경영대학원 경영정보학과(경영학석사)
1994년~현재 경북과학대학 컴퓨터미디어계열(웹마스터전공) 조교수
관심분야 : 컴퓨터그래픽, 컴퓨터미디어, 멀티미디어 데이터베이스

김 명 희



e-mail : hee@wdu.ac.kr
1993년 원광대학교 컴퓨터공학과(공학사)
1996년 원광대학교 대학원 컴퓨터공학과(공학석사)
2001년 원광대학교 대학원 컴퓨터공학과(공학박사)
2001년~2002년 원광대학교 전기전자·정보공학부 BK21 계약교수
2002년~현재 원광디지털대학교 컴퓨터정보학과 전임강사
관심분야 : 분산운영체제, 분산 객체모델, 멀티데이터베이스

주 수 종



e-mail : scjoo@wonkwang.ac.kr
1986년 원광대학교 전자계산공학과(학사)
1988년 중앙대학교 대학원 컴퓨터공학과(공학석사)
1992년 중앙대학교 대학원 컴퓨터공학과(공학박사)
1993년~1994년 미국 Univ. of Massachusetts at Amherst, 전기 및 컴퓨터공학과, Post-Doc.
2002년~2003년 미국 Univ. of California at Irvine 전기공학 및 컴퓨터과학과 연구교수
1990년~현재 원광대학교 컴퓨터공학과 교수
관심분야 : 분산실시간 컴퓨팅, 분산 객체 모델, 시스템 최적화, 멀티미디어 데이터베이스