

표준 인터페이스를 이용한 데이터베이스 통합

권도훈*, 박성공, 이정옥, 백두권
고려대학교 컴퓨터학과 소프트웨어 시스템 연구실
e-mail: {dhkwon, skpark, ljo, baik}@software.korea.ac.kr

An Integration of Databases Using Standardized Interfaces

Do-Hoon Kwon*, Sung-Kong Park, Jeong-Oog Lee, Doo-Kwon Baik
Software System Lab, Dept. of Computer Science & Engineering, Korea University
요 약

온라인 정보 소스와 정보 사용자의 수는 인터넷과 통신 기술의 확산으로 급속히 증가하고 있다. 이는 정보시스템간의 통합을 요구한다. 정보 시스템 통합에 대한 연구는 오래 전부터 시작되었고, 많은 결과들이 발표되고 있다. 하지만 기존의 정보통합 방법은 일반적으로 CDM(Common Data Model)작성과 스키마간 통합 등을 요구하고 있다. 이는 최종 사용자에게 다양한 질의를 제공하기 위함이다. 하지만, 일반적으로 시스템 최종사용자는 사용자 인터페이스 수준의 질의 유형만을 요구한다. 이 경우 다양한 질의를 제공하기 위한 기존 정보시스템들의 요구사항은 많은 오버헤드로 작용한다. 본 논문은 웹환경과 같이 질의 유형이 제한된 곳에서 정보시스템 통합 시 기존 정보 통합 방법의 오버헤드를 제거하기 위한 방법으로서 데이터베이스 접근 인터페이스를 중심으로 한 정보 통합 방법을 제안한다. 이는 질의유형이 제한된 곳에서 기존의 데이터베이스 통합 방법에 비해 단순하고, 확장성 있는 데이터베이스 통합을 제공한다.

1. 서 론

온라인 정보 소스와 정보 사용자의 수는 인터넷과 통신 기술의 확산으로 급속히 증가하고 있다. 이는 정보시스템간의 통합을 요구한다. 정보 시스템 통합에는 정보 시스템간 이질성의 문제가 발생한다. 보통 정보 시스템 이질성을 크게 시스템과 정보의 이질성으로 나누고 있다[1]. 이중 시스템 이질성은 많이 해결이 되는데 비해, 정보 이질성 문제는 연구가 진행중이다 [1]. 지금까지 정보의 이질성 문제를 해결하기 위해 제안된 방법들은 비슷한 유형의 모델을 바탕으로 하고 있다. 즉, 일반적으로 정보 통합시스템은 CDM(Common Data Model)을 갖추고 있으며, 사용자 서비스 요청(질의)을 해결하기 위한 부질의(sub-query) 집합으로의 번역 등 복잡한 구조를 기반으로 하고 있다. 이는 최종 사용자에게 다양한 질의를 제공하기 위함이다. 하지만 질의의 유형이 사용자 인터페이스에 제한되는 도메인에서는 이런 접근은 큰 오버헤드가 될 수 있다. 본 논문에서는 웹 기반 응용프로그램과 같이 질의의 유형이 제한된 도메인에서 데이터베이스 접근에 관련된 코드들을 함수로 선언하고 이들을 인터페이스(선언된 함수들의 집합)로 묶어 CDM, 부질의 번역, 지역 데이터베이스 스키마와 CDM 스키마 간 번역의 사용을 배제하여, 간단하며 확장이 용이한 데이터베이스 통합 방법을 제안한다. 본 논문에서 제안한 데이터베이스 통합 방법은 질의의 유형이 유사하고 질의의 수가 제한된 전자상거래 등에서 효과적이다. 이런 도메인에서 보통 사용자들은 사용자 인터페이스를 통해 일반적으로 정해진 서비스만을 요청할 수 있기 때문에, 결국 질의의 유형이 한정될 수 있다.

2. 관련 연구

지금까지 여러 연구에서 분산되며, 이질적이고, 동적인 정보 시스템을 통합하는 방법들을 연구하고 있다. 이들은 일반적으로 CDM과 부질의의 작성이라는 공통된 작업을 요구한다. 먼저 다양한 정보 시스템 통합 방법을 6가지 유형으로 살펴보고, 정보 시스템 통합 방법의 일반적 모델을 기술한다.

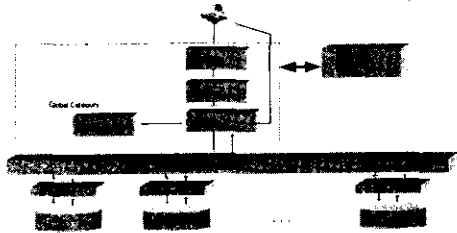
2.1 다양한 정보 시스템 통합 방법

첫째, 고전적 접근 방식이 있다. 이 방식은 다양한 지역 소스(local source) 스키마의 차이를 해결하기 위한 하나의 전역적 스키마 작성이 핵심이 된다. 즉, 이질적인 지역소스의 스키마를 바탕으로 전역 스키마를 작성하고, 이를 통해 시스템 사용자는 다양한 지역 소스에 대해 일정한 형태(전역 스키마)로의 접근이 가능하며, 다양한 지역 소스가 통합되는 효과를 준다. 하지만 지역소스가 새로이 추가되거나 수정되면 지역 소스를 바탕으로 한 전역 스키마의 갱신이 요구되는 단점이 있다 [2,3]. 둘째, 고전적 접근방식이 유효했던 데이터베이스 사용자 이질성의 문제를 해결한 연합된 접근 방법(federated approach)이 있다. 하지만 여전히 전역 스키마의 방식을 사용하고 있고, 전역 스키마 또한 여전히 정적이기 때문에 지역 소스의 추가나 수정 시 전역 스키마의 갱신이 요구되어진다 [2,4]. 셋째, 분산 객체 관리방식(distribute object management approach)이 있다. 이는 분산되고 이질적인 데이터베이스를 분산 객체 공간의 객체집합 모델 기반으로, 연합된 접근 방법을 일반화 시켰다. 이것은 공통 객체 모델과 공통 객체 질의 언어를 바탕으로 하고 있다[2,5]. 넷째, 지능적 정보 통합 방식(intelligent information integration(I3))이 있다. 이것은 3계층 구조를 바탕으로 한 중계자(mediator) 아키텍처를 사용한다 [2,6]. 중계자는 일반적으로 지식을 기반으로 데이터베이스 스키마 정보를 가지고 있으며, 룰(rule)을 통하여 서로 다른 데이터베이스의 내용을 통합할 수 있다[7]. 다섯째, 온톨로지(ontology) 기반의 정보 통합 방식이 있다. 온톨로지는 일반적으로 특정 도메인의 개념과 그들의 관계를 정의해 놓은 지식 베이스이다. 이들은 실제 소스들과 독립적으로 개발되기 때문에 지역 소스의 추가, 삭제, 수정 등에 유연하다[11]. 마지막으로 자율성(autonomy), 지능성(intelligence), 이동성(mobility), 사교성(social ability) 등의 특성을 가진 에이전트 기반의 정보통합 방법이 있다[8].

2.2 데이터베이스통합 방법의 일반적 모델

<그림1>은 FIB 모델[9]와 [10],[11]에서 제안한 모델을 바탕으로 한 데이터베이스 통합의 일반적 모델이다. 일반적으로 이들은 CDM, 사용자 질의의 부질의 집합으로의 번역(mediator),

각각 생성된 부질의에대한 수행순서의 조절(optimizer), 부질의의 해당 지역 소스로 전달(executioner), 각 지역소스의 정보를 가지는 데이터베이스(global category), 각각의 부질을 물 지역 소스 언어로 번역(wrapper), 그리고 모델에서 제공하는 질의 결과의 처리(executioner) 등을 요구하고 있다.



<그림 1> 데이터베이스통합 방법의 일반적 모델

3. 표준 인터페이스를 이용한 데이터베이스 통합방법

전자상거래와 같은 웹 기반 응용프로그램의 최종 사용자는 사용자 인터페이스를 통하여 작업을 하게 된다. 이런 시스템들의 질의 유형은 보통 사용자 인터페이스에서 제공하는 서비스의 수와 거의 유사하게 제한적이다. 일반적인 도메인의 경우도 유사하다. 동일 도메인에서 제작된 응용프로그램들은 보통 비슷한 일을 처리하고 있다. 결국, 유사한 사용자 인터페이스를 가진다. 이것은 응용프로그램 수준에서 유사한 정보를 필요로 한다는 것을 의미한다. 이런 특징을 바탕으로 우리는 한 도메인에서 한정된 질의만을 필요로 할 수 있고, 그 것을 일반화시킬 수가 있다. 본 논문에서는 이러한 일반적이고 제한된 질의들을 동일 도메인에서 구축된 응용프로그램들을 통합하는데 이용한다. 한편 웹이나 동일 도메인에서 구축된 데이터베이스 기반 응용프로그램을 통합하기 위해, 기존에 사용하게 되는 데이터베이스 통합 방법을 사용한다면 CDM 작성과 스키마 통합에 관련된 많은 작업을 요구한다. 하지만 본 논문에서는 위에서 언급한 일반화된 질의를 바탕으로 인터페이스를 작성하고 이를 기반으로 데이터베이스 통합을 하면서 기존 데이터베이스 통합의 많은 오버헤드를 제거한다. 다음은 본 논문에서 제안한 데이터베이스 통합 모델의 아키텍처를 기술한다. 그리고 본 모델을 단순화 시켜 동작원리를 설명하고, 마지막으로 이 모델을 사용하기 위한 시나리오를 기술한다.

3.1 데이터베이스 통합 모델

기존의 단일 데이터베이스 시스템은 최근 분산 환경(클라이언트/서버환경)의 보급과 함께 응용프로그램층(비즈니스 로직과 사용자 인터페이스로 구성), 네트워크 층, 소스 층으로 구성된 3개 층 아키텍처로 구성되어있다. 본 표준 인터페이스를 이용한 데이터베이스 통합 모델은 이 아키텍처를 다시 5개의 층으로 확장, 데이터베이스 통합을 할 수 있도록 했다. <그림2>에서와 같이 데이터베이스 통합 모델은 응용프로그램(application) 층, 통합(integration) 층, 네트워크 층, 인터페이스 층, 지역 소스 층으로 확장되었다.

• 응용프로그램 층(application layer)

기존의 단일 데이터베이스 모델 시스템의 응용프로그램과는 조금 차이가 있다. 보통 단일 데이터베이스 모델 시스템은 데이터베이스 관련 코드들이 응용 프로그램층(비즈니스 로직과 사용자 인터페이스로 구성)에 공존하지만, 본 모델에서는 데이터베이스 접근 코드들을 인터페이스 층에서 별도로 관리하게 된다.

• 통합 층(integration layer)

본 논문에서 제안하는 데이터베이스통합 모델은 다수의 데이터베이스를 통합하는 것이 목적이다. 그래서 단일 데이터베이스 시스템과는 달리 몇 가지 추가되는 기능들이 요구된다. 우선 하나의 서비스 요청은 다수의 질의들을 요구할 수도 있다. 이때, 생성된 다양한 질의들은 병렬적 관계일 수도 있고, 인과적 관계에 있을 수도 있다. 즉, 생성된 질의들 중 어떤 질의는 다른 질의의 결과를 바탕으로 생성될 수도 있는 것이다. 이처럼 생성된 질의들에 대한 스케줄링이 필요한데 이것을 최적화기(optimizer)가 처리한다. 다음 다수의 데이터베이스를 기반으로 하기 때문에 각각의 질의들은 적절한 지역 데이터베이스로 보내져야 하는데 이 작업을 처리기(executioner)가 처리하게 된다. 이때, 각 지역 데이터베이스와 관련된 정보는 지역 데이터베이스관리자(global category)를 바탕으로 한다[9].

• 네트워크 층(network layer)

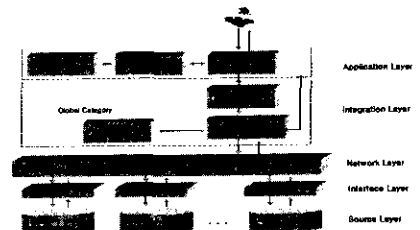
본 시스템은 분산된 환경에서의 데이터베이스를 기반으로 하고 있다. 이와 같은 분산환경 처리를 위해 자바의 RMI(Remote Method Invocation)를 이용한다.

• 인터페이스 층(interface layer)

응용프로그램층에서 분리된 데이터베이스 접근을 위해 선언된 함수들의 집합이다. 이는 적용 도메인별로 다르게 정의가 된다. 인터페이스를 구성하는 함수들은 해당 도메인의 일반적 정보를 요구하게 된다.

• 소스 층(source layer)

실제 각각의 지역 데이터베이스를 의미한다. 이들 각각의 지역 데이터베이스는 동일한 도메인에 대하여 다양한 스키마를 바탕으로 데이터베이스를 구축하고 있다. 다양한 데이터베이스 통합 시 이들 각각의 데이터베이스를 바탕으로 인터페이스 층의 인터페이스 내 함수들을 구현한다.



<그림 2> 표준 인터페이스를 통한 데이터베이스통합모델

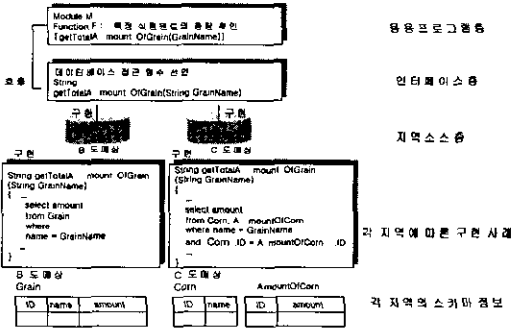
3.2 데이터베이스 통합 모델의 동작 원리

본 논문에서 제안한 데이터베이스 통합 모델의 효과적 이해를 위해서 앞서 소개한 5개의 층을 응용프로그램 층, 인터페이스 층, 지역소스 층으로 단순화 시켜 동작원리를 기술한다. 동일 도메인에서 구축된 데이터베이스 기반 응용프로그램들은 비슷한 서비스를 제공하며, 동일한 정보를 필요로 한다. 본 논문에서 제안하는 데이터베이스 통합 모델은 한 도메인 내에서 구축된 단일 데이터베이스기반 응용프로그램들이 사용하는 질의들 중 일반적인 것들을 추출하여 함수로 선언하고, 응용프로그램의 비즈니스 로직에서 분리해서 인터페이스로 묶는다. 그리고, 논문에서 제안한 모델에 통합되는 지역 소스들은 자신의 데이터베이스 스키마에 적합하게 앞에서 작성한 인터페이스를 구현하게 된다. 이로서 기존의 단일 데이터베이스 응용프로그램이 비즈니스 로직에서 직접 데이터베이스를 접근한 것과 달리, 다양한 정보들에 대한 함수들을 정의해둔 인터페이스를 통해 데이터베이스에 접근하게 되면서 데이터베이스 통합이 가능하게 된다. 전체적인 이해를 위해 SCM(Supply Chain

Management) 기반 응용프로그램 예를 본다. A라는 기업은 다양한 식품을 판매하는 기업이다. A기업은 B와C라는 곡물 도매상으로부터 식품원료를 제공받는다. 그리고 곡물을 제공하는 도매상들은 계속증가 되며, 각각이 자체적인 데이터베이스가 구축되어있다. 이때, 기업A는 도매 업체들로부터 얻을 수 있는 식품원료의 총량(보유량)을 각 종류별로 알 필요가 있다. <그림3>은 A라는 기업이 각 도매업체로부터 제공받을 수 있는 식품원료의 총량을 얻는 경우의 예로서, 인터페이스는 다음과 같은 함수를 사용해서 선언한다.

`int getTotalNumberOfGrain(String GrainName)`

응용프로그램층(비즈니스 로직과 사용자 인터페이스로 구성)에서 식품원료의 양을 필요로 하는 모델들은, 단순히 `getTotalNumberOfGrain()` 함수를 호출하게 된다. 한편 이 시스템에 통합하고자 하는 지역 소스들은 스스로 이 인터페이스에 대해서 각자 지역 스키마에 적합하게 구현을 해주게 된다. <그림3>에 “각 지역에 따른 구현 사례”의 구현 박스 부분을 참조하면 쉽게 이해할 수가 있다. 특정 도메인에서 일반적으로 사용되는 이와 같은 정보들을 바탕으로 인터페이스를 작성하고, 이를 통해 데이터베이스통합을 수행한다.



<그림3> 데이터베이스통합모델의 동작 원리

3.3 데이터베이스 통합 모델의 구축 시나리오

본 논문에서 제시하는 모델을 통해 데이터베이스를 통합하고자 하는 경우 다음과 같은 절차를 통해 데이터베이스 통합을 수행한다.

- 먼저 적용될 도메인의 특성을 관찰하고 본 시스템의 적용대상인지를 분석한다. 즉, 질의의 패턴이 일반화될 수 있는 도메인인가를 확인한다.
- 다음 작업 도메인을 기반으로 구축된 응용프로그램을 분석함으로써, 현 도메인에서 사용되는 일반적인 데이터를 식별한다.
- 식별된 데이터를 기반으로 각각에 대한 함수를 선언한다.
- 선언된 함수들을 바탕으로 인터페이스를 작성한다.
- 이렇게 작성된 인터페이스를 기반으로 데이터베이스 통합 시스템을 구축한다.
- 새로운 데이터베이스 추가 시 추가되는 데이터베이스의 관리자는 제공되는 인터페이스를 <그림3>에 구현부분과 같이 자신의 데이터베이스 스키마에 맞게 구현한다.

4. 기존 데이터베이스통합모델과의 비교

기존의 일반적 데이터베이스 통합 모델과 본 논문에서 제안한 모델의 장단점을 비교하면 다음과 같다. 먼저 장점은 다음과 같다. 우선, 시스템의 구조가 간단하다. <그림2>와 <그림1>을 통해 둘 간을 비교해 보면 본 논문에서 제안하는 시스템에는 CDM이 없고, 중계기 기능이 없으며, 해당 질의를 지역 소스로 번역하는 맵퍼 대신 데이터베이스 접근 관련 함수들의 집합인 인터페이스로 교체되었다. 이는 시스템을 훨씬 단순화시켰

으며 구축하기도 많이 쉬워진다는 장점을 가지고 있다. 또한 스키마나 각각 데이터베이스 요소의 의미들에 대한 지능적 처리를 인터페이스 구현을 통해 단순화 시켰다. 둘째, 데이터베이스의 확장성이 높다. 제안된 인터페이스를 각 지역 데이터베이스에서 지원만 한다면 어떤 데이터 모델에 상관없이, 혹은 스키마 구조에 상관없이 확장이 가능하다.

단점은 다음과 같다. 우선 통합되어 질 지역 데이터베이스는 항상 제공한 인터페이스를 구현해야 한다. 즉, 통합 시 지역 데이터베이스 쪽의 오버헤드가 크다 둘째, 질의의 수, 즉 인터페이스에서 선언하는 함수의 수가 제한된다. 보통 인터페이스의 규모를 크게 하면 정보 통합 모델을 좀더 일반화 할 수 있지만, 지역 소스쪽의 부담이 더해진다. 셋째, 주어진 도메인에서 일반적인 정보처리밖에 할 수가 없다. 즉, 특정 지역 소스에서 제공했던 모든 정보를 받아볼 수 없다. 이는 미리 인터페이스를 정의하고 그것을 바탕으로 정보통합 시스템을 구축하기 때문에 발생하는 단점이다. 그러나 질의의 수 제한의 문제는 일반적으로 응용프로그램 수준에서 최종 사용자는 다양한 질의보다는 사용자 인터페이스를 통한 한정된 질의만을 사용한다는 점에서 크게 문제가 되지 않는다.

5. 결론 및 향후 연구 과제

본 논문에서는 질의의 유형이 적은 도메인에서 보다 쉽게 데이터베이스를 통합하는 방법을 제안했다. 이것은 JDBC(Java Database Connectivity)나 ODBC(Open Database Connectivity)가 시스템 레벨의 데이터베이스 통합을 해결한 것과 유사하게 특정 도메인에서 일반적으로 사용되는 질의들을 바탕으로 데이터베이스 관련 인터페이스를 작성하고 그것을 바탕으로 스키마와 의미 레벨의 데이터베이스 통합을 하고 있다. 이는 제한된 질의의 유형을 가지는 곳에서 기존의 데이터베이스통합 모델에 비해 간단하고 확장성 있게 데이터베이스 통합을 수행한다.

본 논문에서 제안한 시스템의 모델을 바탕으로 추후에 구현을 할 예정이다. 또한 현재 제안된 모델은 사용되는 질의의 유형이 유한한 곳에서만 적용이 가능한데, 향후에는 본 데이터베이스통합 모델이 다양한 질의를 서비스 할 수 있는, 즉 일반적으로 적용될 수 있는 방법으로서의 확장에 대한 연구도 필요하다.

<참고 문헌>

- Aris M. Ouksel, Amit P. Sheth: Semantic Interoperability in Global Information Systems: A Brief Introduction to the Research Area and the Special Section. SIGMOD Record 28(1): 5-12, 1999.
- Maurizio Panti, Luca Spalazzi, and Alberto Giretti, "A Case-Based Approach to Information Integration", Proceedings of the 26th VLDM Conference, Cairo, Egypt, 2000.
- S.Ram, "Special issue on heterogeneous distributed database systems", IEEE Computer Magazine, 24(12), December 1991.
- A. Sheth and J. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", ACM Transaction on Database Systems, 22(3), 1990.
- T.Ozsu, U.Dayal, and P.Valduriez, "Distributed Object Management", Morgan Kaufmann, San Mateo, CA, 1993.
- G. Wiederhold, "Mediators in the Architecture of Future Information Systems", IEEE Computer Magazine, 25:38-49, March 1992.
- 이강찬, 김덕현, 이규철, "차세대 데이터베이스 통합모델에 관한 연구", 98춘계학술발표논문집, 정보과학회, 1998.
- K. Decker and K.P Sycara, "Intelligent adaptive information agents", Journal of Intelligent Information Systems, 9(3):239-260, 1997.
- Evangelia Pitoura, Omran Bukhres, and Ahmed Elmagarmid, "Object Oriented in Multidatabase Systems", Purdue University, ACM, 1995.
- Soon W. Chung and Pyeong S. Mah "Schema Integration for Multidatabases Using the Unified Relational and Object-Oriented Model", ACM 0-89791-737-5, 1995.
- Jerry Fowler, Brad Perry, Marian Nodine, and Bruce Bargmeyer, "Agent-Based Semantic Interoperability in InfoSleuth", SIGMOD Record 28(1): 60-67, 1999.