

데이터 통합 시 충돌 문제를 해결하는 미디어이터의 설계 및 구현[†]

이경하⁰, 김정일, 김태현, 이강찬, 이규철
충남대학교 컴퓨터공학과 데이터베이스 연구실
{bart, kyung1, thkim, dolphin, kcllee}@ce.cnu.ac.kr

Design and Implementation of A Mediator System for Resolving Data Conflict on Data Integration

Kyong-Ha Lee⁰, Kyung-Il Kim, Tae-Hyun Kim, Kang-Chan Lee, Kyu-Chul Lee
Dept. of Computer Engineering, Chungnam National University

요 약

인터넷과 같은 정보통신망의 발전에 따라 기존에 구축되었던 정보 자원들은 사용자들의 접근이 보다 빈번해졌으며, 접근의 유용함 때문에 기존보다 더 많은 정보 자원들이 새롭게 구축되고 있는 실정이다. 이러한 정보 자원들은 각기 상이한 목적으로 구축되어 사용자가 요구하는 정보를 얻기 위해서는 여러 정보 자원들에 빈번히 접속해야 하는 불편함이 증가하고 있다. 이를 해결하기 위해 최근에 들어 상이한 정보 자원들을 통합하고자 하는 연구가 진행되고 있다. 정보 자원들의 통합에 있어서 가장 큰 문제점은 서로 이질적인 데이터들을 통합할 때의 데이터 충돌을 극복하면서도 각 정보 자원들의 자치성들을 보장할 수 있어야 한다는 점이다. 본 논문에서는 정보 자원 통합 시 발생하는 데이터 충돌을 해결하면서 각 정보 자원들의 자치성을 보장하는 미디어이터를 이용한 통합 시스템을 설계, 구현함으로써 이런 문제점을 해결하는 방안을 제시한다.

1. 서론

인터넷의 급속한 확산에 따라, 인터넷사용자가 급격히 증가하기 시작했으며, 이에 따라 사용자들이 정보 시스템에 접근하는 일이 더욱 빈번해지고 더 많은 정보를 요구하게 되었다. 현재의 정보 자원들은 이들이 요구하는 데이터를 제공하기 위해 그 수가 더더욱 증가하는 추세이다.

이러한 정보 자원들의 데이터를 요구하는 사용자들은 서로 다른 정보 자원들을 필요에 따라 빈번하게 접근하고자 한다. 그러나, 정보 자원들은 각기 상이한 목적으로 서로 다른 시스템과 데이터 모델들을 사용하면서 지리적으로 분산되어 구축되어 왔으므로, 각 시스템들의 자율성을 보장하면서 통합된 서비스를 제공하기란 쉬운 일이 아니다. 최근에 들어 이러한 어려움을 덜기 위해 분산되고 이질적인 정보 자원들의 통합에 대한 연구가 진행되고 있으며, 이러한 연구들은 크게 연합 데이터베이스, 데이터 웨어하우스, 멀티데이터베이스, 미디어이터를 이용한 데이터 통합 방법[1]을 기초로 수행되고 있으며, 각 시스템들의 장단점은 [1]에 나타나 있다. 본 논문에서는 연구 개발된 정보 자원 통합 시스템

(HyDIM Mediator)[2]을 통하여 미디어이터 방식을 이용한 통합 방법과 구현한 시스템에서의 데이터 통합 시 발생하는 충돌 문제를 분류, 해결하는 방법에 대하여 기술한다.

2. 시스템 구성

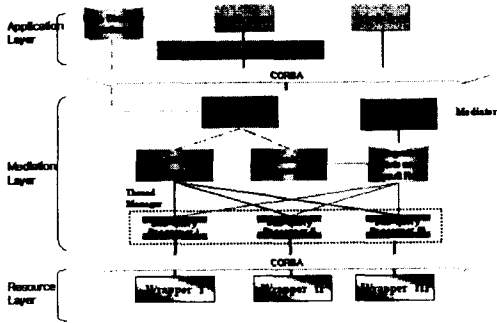
본 연구에서 설계한 미디어이터 모델은 CORBA 환경[3]과 룰을 기반으로 분산적이고 이질적인 정보 자원들을 통합[4]하는 방법을 제공한다. 구성요소로는 ORB(Object Request Broker), 미디어이터, 래퍼, 사용자 인터페이스로 이루어져 있다. 미디어이터는 룰 기반의 정보 통합을 제공하며, 글로벌 스키마 관리기, 질의 분해기, 동적 룰 생성기, 서브 질의 처리기로 구성된다. 이 구조를 도식화하면 [그림 1]과 같다.

[그림 1]에서 보는 바와 같이 본 시스템은 어플리케이션 계층, 미디어이션 계층, 자원 계층으로 이루어지며 각각의 기능은 다음과 같다.

■ 미디어이션 계층(Mediation Layer)

클라이언트 어플리케이션이나 글로벌 질의 인터페이스로부터의 글로벌 질의를 입력 받아 각 로컬 스키마에 적당한 형태로 질의 분해 후, 각 서브 질의들을 자원 계층에 전달하고 수행 결과를 통합하는 기능을 수행한다. 또한, 글로벌 스키마와 로컬 스키마를 관리(생성, 수정, 삭제)하는 기능을 수행한다.

[†]본 논문은 국방과학연구소의 분산이종DB연동기술연구(과제호:UD980002GD)의 위탁으로 수행된 과제임



[그림 1] 시스템 구조도

■ 자원 계층(Resource Layer)

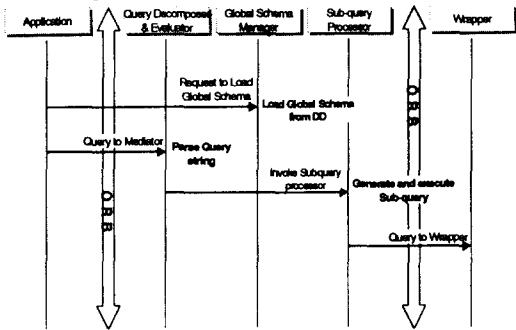
미디어이터로부터 서브 질의를 전달 받아 각 정보 자원에 적합한 로컬 질의로 변환 수행하고, 그 결과를 다시 미디어이터에 전달하는 기능을 수행한다.

■ 어플리케이션 계층

미디어이션 계층에 대한 인터페이스와 관리 도구 등을 포함하며 본 시스템이 제공하는 API를 이용하여 작성된다.

3. 질의 처리와 글로벌 스키마

사용자가 어플리케이션 계층을 통해 전달한 글로벌 질의는 미디어이션 계층에서 처리된다. 글로벌 질의의 처리는 글로벌 스키마의 로딩 과정과 실제 글로벌 질의 처리 과정 두 단계로 나누어지는데, 다음은 그 과정에 대한 설명이다.

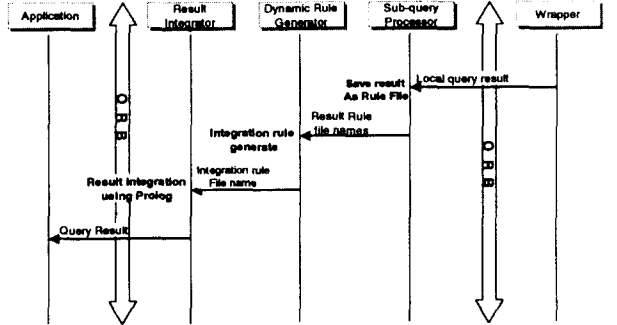


[그림 2] 글로벌 질의 처리 과정

- 1) 질의 수행에 앞서 질의 대상이 되는 글로벌 스키마를 로딩토록 한다.
- 2) 질의 분해기는 전달 받은 질의를 파싱하여 파싱 트리(Parsing Tree)를 생성하며, 글로벌 스키마를 참조하여 해당 서브 질의의 처리기들을 생성한다.
- 3) 각 서브 질의의 처리기들은 전달 받은 파싱 트리를 로컬 스키마에 적절한 형태로 변형하여 해당 래퍼에 대한 질의로 변환 후 서브 질의를 ORB를 통해 래퍼로 전달하여 실행케 한다.

질의 결과의 통합은 각 서브 질의들의 결과를 글로벌 스키마를 기반으로 스키마의 충돌과 결과의 중복을 제거하여 이루어지며 결과 통합기에서 이 작업을

수행한다. 동적 룰 생성기는 글로벌 스키마와 글로벌 질의 분해 시 만들어진 질의 파싱 트리를 이용하여 통합 룰을 자동으로 생성하며, 서브 질의의 결과로부터 얻어진 결과를 룰로 변환하여 결과 통합기에서 최종 질의 결과를 생성한다. 질의 결과의 통합 과정은 [그림 3]과 같다.



[그림 3] 질의 결과의 통합 과정

본 통합 시스템은 글로벌 스키마를 선언적으로 표현하기 위하여 혼 클로уз드를 사용한다. 혼 클로уз드로 표현된 글로벌 스키마에는 스키마 정보 뿐만 아니라 로컬 스키마와의 관계, 통합 시 발생할 수 있는 데이터 충돌을 해결하기 위한 외부 함수의 호출, 검색 조건들이 포함된다. [그림 4]는 글로벌 스키마를 이용하여 각 정보 자원이 수행한 질의 결과를 통합하는 예이다.

```

result(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL) :-
consult( '/var/tmp/aaa00u7zm.P' ),
consult( '/var/tmp/baa00uzze.P' ),
consult( '/var/tmp/caa01uk2x.P' ),
result_0(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL).

result_0(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL) :-
result_1(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL),
result_2(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL).

result_1(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL) :-
ware_1(PRODUCT_NAME, PRICE),
step_1(PRODUCT_ID, PRODUCT_NAME, SUBCATEGORY, SHAPE),
xml_1(PRODUCT_ID, PRODUCT_NAME, TITLE, MANUAL).

result_2(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL) :-
PRODUCT_ID=_, PRODUCT_NAME=_, PRICE=_,
PRODUCT_ID=_, PRODUCT_NAME=_, SUBCATEGORY=_, SHAPE=_,
xml_2(PRODUCT_ID, PRODUCT_NAME, TITLE, MANUAL).
    
```

[그림 4] 글로벌 스키마와 결과 통합 룰

위의 예에서 result는 글로벌 스키마를 바탕으로 사용자 질의에 따라 만들어진 하나의 뷰(view)이며, consult()와 result_0 을 통하여 얻어진다. result_0 은 다시 result_1과 result_2를 수행한 결과로 AND 연산으로 합쳐지게 되며, 각 result_1,2는 ware, step, xml 절로 이루어져 있는데, 이는 실제 통합하고자 하는 정보 자원들이 미디어이터에 반환한 질의 결과의 형식이다.

4. 데이터 통합 시의 충돌 해결

4.1 충돌의 분류

각 정보 자원들의 데이터 통합 시 각 데이터들 간의 충돌(Conflict)이 발생할 수 있다. 이러한 데이터들 간의 충돌은 데이터 통합 과정을 어렵게 하는 중요 원인이며, 각 정보 자원들이 저장하고 있는 데이터의 데이터 모델, 스키마의 이질성에 의하여 발생한다. 다음은 정보 통합 시 발생할 수 있는 충돌의 분류[5]이다.

■ 스키마 충돌(Schema Conflict)

스키마 충돌은 각 정보 자원들이 보유한 데이터들의 스키마 이질성에 의해 발생하며 Naming conflict와 Structural conflict가 이에 해당한다. 엔티티, 어트리뷰트 간의 이름의 상이함으로 발생하는 것이 Naming Conflict이며, Structural conflict는 통합할 정보 자원들의 스키마 구조가 다른 경우, 즉 정보 자원 1에서는 엔티티인 데이터가 정보 자원 2에서는 어트리뷰트 등으로 사용되는 경우가 이에 해당된다.

■ 데이터 충돌(Data Conflict)

데이터 충돌은 스키마에 저장되어 있는 데이터간의 이질성에 의해 발생하는 충돌이다. 데이터 타입의 불일치로 발생하는 type conflict, 데이터의 Unit이 다름으로써 발생하는 Measurement conflict, measurement conflict와 유사하나 데이터의 단위가 서로 포함관계가 있을 경우, 즉 'monthly pay'와 'yearly pay'의 예처럼 데이터의 Unit이 서로 다른 경우에 발생하는 Granularity conflict, 동일한 의미의 데이터를 달리 표현하는 경우(학성의 성적을 'A+'와 'Excellent'로 표현하는 경우 등)에 해당하는 Representation Conflict 등이 데이터 충돌에 해당한다.

4.2 데이터 통합 시 충돌의 해결

본 논문에서 소개하는 정보 자원 통합 시스템은 위에서 분류된 데이터 충돌을 다음과 같은 방법으로 해결하고 있다.

- 1) 스키마 충돌 중 naming conflict는 글로벌 뷰와 로컬 스키마의 매핑 규칙을 이용하여 해결한다.
- 2) structural conflict의 경우 자원 계층(Resource Layer)에서 해결한 후 미디어데이터에 질의 결과를 파라미터 형식으로 전달함으로써 이를 해결한다.

Multi-Database	STEP	XML
<pre>create class PRODUCT (region, product_id string, product_name string, price monetary, sizes integer, color string, supplier string)</pre>	<pre>SCHEMA config_ctrl_design; TYPE -- END_TYPE; ENTITY product id -- Identifier; name : label; UNIQUE URI : id; END_ENTITY; ENTITY approval_status ... END_ENTITY; END_SCHEMA;</pre>	<pre><?xml version="1.0"?> <?xmlstylesheet ...?> <DOCTYPE --> <MANUAL> <INFO> <TITLE>Guide</TITLE> <PRODUCT> <PID>TW-B-1004</PID> <PNAME>Bridge</PNAME> </PRODUCT> </INFO> <MANUAL></pre>

[그림 5] Structural Conflict의 예

위의 예는 본 시스템이 통합하고자 하는 정보 자원들의 스키마 구조이다. product_id는 warehouse에서는 어트리뷰트로 STEP에서는 identifier로 XML에서는 element로 존재하고 있음을 알 수 있다.

- 3) 데이터 충돌의 경우 본 시스템은 외부 매핑 함수(External Mapping Function)을 호출함으로써 데이터의 Unit을 통합 시에 적절한 Unit으로 매핑 시켜 이를 해결한다. type conflict의 경우 프롤로그에서 처럼 타입을 필요치 않는 중립 데이터 모델을 사용함으로써 충돌을 피하는 방식을 택한다.

[그림 6]은 결과 통합 과정 중 해당 데이터의 표현이 상이함을 해결하고자 등록된 외부함수를 호출하여 이를 해결하는 과정이다.

```
result(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL) :=
consult( '/var/tmp/aaa00uYzm.P' ),
consult( '/var/tmp/baa00uZze.P' ),
consult( '/var/tmp/caa01uKzx.P' ),
consult( map, [cc(gcc), cc_opts("-O2 -/usr/local/X11/R6/include")] );
result_0(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL) :=
result_1(PRODUCT_ID, PRODUCT_NAME, PRICE, SUBCATEGORY, SHAPE, TITLE, MANUAL) :=
ware_1(PRODUCT_ID, PRODUCT_NAME, PRICE),
step_1(PRODUCT_ID, PRODUCT_NAME, X, SHAPE),map(X, SUBCATEGORY);
xml_1(PRODUCT_ID, PRODUCT_NAME, TITLE, MANUAL);

#include "XSB/emu.cintf.h"
void map(void){
  char *A = ptop_string(1);
  if( !strcmp(A, "A+") )
    ctop_string(2, "Excellent" );
}
```

[그림 6] Representation 충돌의 해결 예

위의 그림은 STEP 정보자원에서 SUBCATEGORY 어트리뷰트의 표현이 상이함으로써 발생하는 충돌을 해결기 위하여 MAP 절을 외부함수로서 프롤로그 엔진에 등록한 후, 결과 통합 시에 이를 호출하여 'A+' 이라는 데이터 표현을 'Excellent'로 바꾸는 예이다. Measurement 또는 Granularity에 대한 충돌도 외부 함수로 해결할 수 있으나, 이 경우, 계산 가능한 데이터 Unit에 대한 매핑이므로, 작업량이 많이 걸리는 외부 호출에 의한 방식보다는 프롤로그에서의 기본 연산자들을 사용하여 처리한다.

5. 결론

연구에서 개발된 시스템은 CORBA환경 위에 혼 클로уз드를 이용함으로써 기존의 시스템들의 가지고 있던 복잡성 문제나 시스템 성능 문제를 극복하면서 간결한 통합 정보의 표현으로 데이터 통합이 가능토록 하였다. 미디어데이터 방식을 이용한 정보 통합 시스템은 통합의 대상이 DB로만 한정되지 않고 어플리케이션에서의 통합 또한 가능하므로, 상이한 구조의 시스템들을 통합하는데 용이하다. 이러한 특징은, 구조화된 스키마를 가지는 일반 DB의 통합 뿐만 아니라 전자 상거래, 디지털 라이브러리 분야에도 적용이 가능하며 향후 EAI(Enterprise Application Integration) 환경에도 중요한 역할을 할 것이라 기대된다.

6. 참고 문헌

- [1] H. Garcia -Molina, J. Ullman, J. Widom "Data Integration", Database System Implementation, Ch 11, P.595-612
- [2] 이경하, 이상, 이강찬, 이규철, "물 기반 미디어데이터의 설계 및 구현", 정보과학회 '99 봄 학술대회논문집(B) P.185-187
- [3] 박재현, "코아 코바", 영한 출판사, 1998.1
- [4] J. Ullman, "Information Integration with Logical View", ICOT '97, 6th International Conference Proceedings
- [5] NIIIP Reference Architecture, <http://www.niiip.org/public-forum/index-ref-arch.html>, NIIIP Consortium