

eMRA: Extension of MRA Considering the Relationships Between MDR Concepts

Young-min Joo[†] · Jangwon Kim^{††} · Dongwon Jeong^{†††} · Doo-Kwon Baik^{††††}

ABSTRACT

Metadata registry (MDR) is the international standard, developed by ISO/IEC for exchange and sharing data between databases. Many MDR systems are used in diverse domains such as medical service, bibliography, environment for sharing and integrating data. However, those systems have different physical structures individually because the MDR standard defines only the metamodel for registering and storing metadata. It causes heterogeneity between the system structures and requires additional cost to maintain interoperability. ISO/IEC 13249-8 Metadata Registry Access (MRA) is developing as an international standard to provide a consistent access facility to data stored in different metadata registries. However, MRA does not consider the relationships between the concepts (classes) defined in the MDR specification. It causes that incorrect query results returned from MDR systems. It also requires additional cost of modeling and rewriting queries to reflect each physical model. Therefore, this paper suggests eMRA which considers the relationships between the concepts in MDR. The comparative evaluations are described to show the advantages of eMRA. eMRA has superior performance in query modeling and referential integrity than MRA defined by the relationship between the concept of MDR.

Keywords : Metadata Registry, ISO/IEC 13249-8, MRA, SQL, Standard Interface

eMRA: MDR의 개념간 관계성을 고려한 MRA 확장

주영민[†] · 김장원^{††} · 정동원^{†††} · 백두권^{††††}

요약

메타데이터 레지스트리(Metadata Registry, MDR)는 데이터베이스 간 데이터 교환 및 공유를 위해 ISO/IEC에서 개발한 국제 표준이며, 의료 서비스, 서지, 환경 등 다양한 도메인에서 데이터 공유 및 통합을 위해 사용되고 있다. 그러나 MDR 표준은 메타데이터 등록 및 저장에 위한 메타모델만을 정의하고 있기 때문에 이 시스템들은 서로 다른 물리적 구조를 가지게 된다. 이로 인해 MDR 시스템 간 불일치가 발생하고 메타데이터의 상호운용을 위해 추가적인 비용이 발생한다. 이러한 문제를 해결하기 위해 ISO/IEC 13249-8 Metadata Registry Access (MRA)가 개발 중에 있으며, MRA는 상이한 MDR 시스템에 일관된 방법으로 접근할 수 있는 표준 인터페이스이다. 그러나 MRA는 MDR 표준에 정의되어 있는 개념 즉, 클래스 간 관계성을 고려하지 않는다. 이는 부정확한 결과를 생성할 수 있으며, 각 MDR 시스템의 물리적 구조를 고려하여 질의를 모델링하고 재작성하는 추가적인 비용이 발생한다. 이 논문에서는 클래스 간 관계성을 고려한 확장된 인터페이스 eMRA(Extended MRA)를 제안하며, 비교 평가를 통해 확장 인터페이스의 장점을 기술한다. eMRA는 MDR의 개념간 관계성을 정의하여 질의 모델링과 시스템의 참조무결성 측면에서 MRA보다 우수한 성능을 가진다.

키워드 : 메타데이터 레지스트리, ISO/IEC 13249-8, MRA, SQL, 표준 인터페이스

1. 서론

메타데이터 레지스트리(Metadata Registry, MDR)는 메타데이터의 등록과 인증을 통하여 표준화된 데이터를 유지 및 관리를 하는 국제 표준이다[1,2]. 또한 메타데이터의 명세 및 의미 공유를 통해 데이터의 상호운용성을 향상시킨다[3-5]. 이러한 MDR의 장점에 기인하여, 현재 다양한 분야에서 MDR 시스템을 개발하여 운영하고 있다[6,7]. 그 중 대표적인 시스템으로는 호주의 보건, 주택, 공동체 서비스를 위한 통계 및 정보에 대한 국가차원의 메타데이터 관리를 위한

* 이 연구에 참여한 연구자는 '2 단계 BK21 사업'의 지원을 받았으며, 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NO.2012-0002439)의 결과물임을 밝히며, 또한 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구인(No.2012M3C4A7033346).

† 준회원: 고려대학교 컴퓨터·전파통신공학과 석사과정

†† 준회원: 고려대학교 컴퓨터·전파통신공학과 박사

††† 종신회원: 군산대학교 정보통계학과 교수

†††† 종신회원: 고려대학교 컴퓨터·전파통신공학과 교수

논문접수: 2012년 9월 11일

수정일: 1차 2012년 11월 8일

심사완료: 2012년 11월 27일

* Co-corresponding Author: Doo-Kwon Baik(baikdk@korea.ac.kr)
Dongwon Jeong(djeong@kunsan.ac.kr)

METeOR[8], 미국의 건강 표준개발기관과 관련 기관 사이의 데이터 요소(Data Element)를 분류하는 USHIK[9] 및 환경 정보에 대한 기록을 표준화하여 사용자 간의 데이터 공유를 가능하게 하는 EDR[10], 암 협회의 caDSR[11], 홈 랜드 보호부와 법무부의 NIEM[12], 법무부의 GJXDM[13], 국회 도서관의 LC[14] 등이 있다. 이 외에도 캐나다의 보건 정보부의 CIHI 데이터 사전[15], 영국의 암 그리드 기관의 암 그리드 MDR[16], 도서관 교육 그리고 문화유산 커뮤니티의 디지털 정보 관리 서비스를 제공하는 UKOLN[17] 등이 있다.

국내에서도 MDR을 기반으로 데이터 통합시스템, 데이터 상호운용, 지리정보, 온톨로지, 서지정보 시스템, 등 다양한 분야에서 연구되고 있으며, 그 중에서도 정보통합 시스템 [18,19]과 상호운용성[20]에 관련한 연구가 주를 이루고 있다.

앞서 언급하였듯이, 다양한 도메인에서 서로 다른 MDR 시스템들이 개발되어 사용되고 있다. 그러나 MDR 표준에서 정의한 개념 모델을 일부 수정하거나 서로 다른 저장 모델을 개발하였다. 또한 각각의 저장 모델을 위한 서로 다른 접근 방법을 사용한다. 이러한 이질적인 구조 및 접근 방법은 서로 다른 MDR 시스템에 저장되어 있는 데이터의 공유 및 재사용을 어렵게 하는 문제점을 야기한다. 또한 데이터 접근을 위한 인터페이스가 각각의 시스템에 종속적이므로, 사용자가 원하는 데이터 요소들을 가져오기 위해서는 각각의 인터페이스 구조를 파악해야 하는 오버헤드가 발생한다[21].

이러한 문제점을 해결하기 위해 ISO/IEC JTC 1/SC 32/WG 4에서는 ISO/IEC 13249-8 Metadata Registry Access (MRA) 표준이 제안되었다[22]. MRA는 MDR에 정의된 클래스에 저장된 데이터에 접근하는 목적을 가진다. 그러나

MRA는 접근에 대한 기본적인 데이터타입만을 제공할 뿐, 구현을 위한 상세한 명세를 제공하지 않는다. 이는 MRA의 실용적인 활용성과 기능성의 한계를 가져온다. 따라서 상세하고 다양한 기능을 제공할 수 있는 질의 인터페이스의 확장이 요구된다. 이 논문에서는 이러한 문제점을 해결하기 위해 MRA 기반의 확장된 질의 인터페이스를 제안한다. 이를 위해 MRA의 기능 확장을 위해 요구되는 사항을 정의하고 이를 해결할 수 있는 방안에 대하여 기술한다. 또한 실제 질의 인터페이스를 구현하고 비교 평가를 통해 제안 모델의 장점을 보인다.

이 논문의 구성은 다음과 같다. 2장에서는 관련연구를 소개하고 3장에서 기존 MRA의 문제점을 기술한다. 4장에서는 이 논문에서 제안하는 MRA 적용방법에 대해 기술한다. 5장에서는 적용한 시스템의 타당성을 검증하기 위한 비교 평가를 하고 6장에서 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

Fig. 1은 MDR의 정보를 이용하기 위한 접근 방법 즉, 직접적인 데이터 접근방법, 어댑터 기반 접근 방법, 바인딩 API 기반 접근 방법, SQL 기반 접근 방법 4가지를 보여준다[8].

Fig. 1의 a)는 MDR 시스템에 직접 접근하는 방법이다. 이 방법은 구축된 시스템의 구조를 정확하게 파악하고 있어야 가능하고, 보안상의 문제가 있기 때문에 일반적으로 사용되지 않는다.

Fig. 1의 b)는 어댑터 기반 접근 방법이다. 이 방법은 사용자와 기관, 상호 간 공동의 어댑터 즉, 인터페이스를 개발

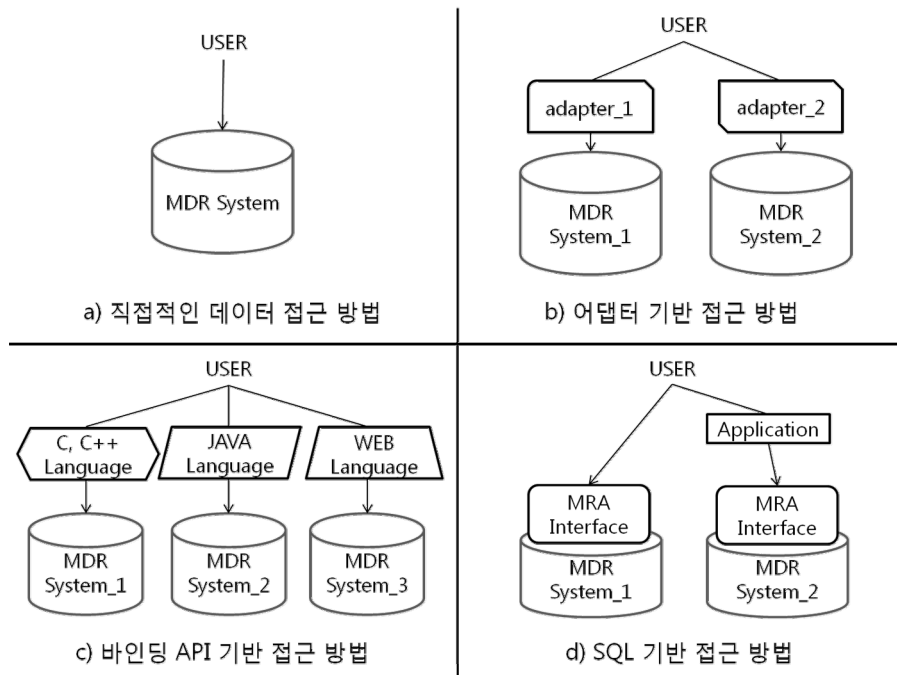


Fig. 1. Four ways to access the MDR system

해야 한다. 그러나 어댑터 기반의 접근 방법은 시스템의 물리적인 구조에 의존적이기 때문에 사용자가 접근하고자 하는 MDR 시스템이 여러 개가 존재할 경우 각각의 시스템에 적용할 수 있는 어댑터를 개발해야 한다. 그러므로 어댑터 개발을 위한 추가적인 비용이 요구된다.

Fig. 1의 c)는 바인딩 API를 이용하여 데이터에 접근하는 방법이다[23,24]. [23]은 C, C++, 자바, 웹 프로그래밍 언어 등과 같은 프로그래밍 언어를 기반으로 MDR 시스템에 접근할 수 있도록 각 언어별로 바인딩 API를 정의한다. 또한 미국 암 센터에서는 웹 프로그래밍 언어를 기반으로 하여, 다양한 MDR에 접근하는 방법을 개발하였다[24]. 이러한 방법들을 통해 MDR 시스템에 접근할 경우, 사용자들은 프로그램 언어에 따라 해당 언어 기반의 API를 파악해야 하므로 추가 비용이 발생한다.

Fig. 1의 d)는 SQL 기반 접근 방법이다. SQL 기반 접근 방법은 메타데이터를 가져오기 위해 SQL을 사용하는 방법이다[25]. SQL은 데이터베이스에 저장되어 있는 데이터를 얻기 위한 표준 질의 언어이다. 따라서 SQL을 이용하면 사용자들이 일관성 있는 문법을 기반으로 데이터에 접근하는 것을 가능하게 한다.

MRA는 앞에서 나열한 4가지 접근 방법들 중에서, 사용자에게 일관성 있는 데이터 접근을 가능하게 하는 SQL 기반의 방법을 적용한다[26]. 특히 MDR에서 정의하고 있는 각각의 클래스들을 SQL에 정의된 사용자 정의 타입(User-defined type, UDT)을 기반으로 정의한다[27,28]. 하지만 기존 연구들은 MDR 표준에 있는 메타 모델들 간의 관계성을 고려하지 않는다. 그 결과 MRA는 표준화된 인터페이스를 지원하기 위해 SQL을 적용하지만, 실제 환경에서 중요한 정보로 활용되는 메타 모델들 간의 관계 정보가 누락될 수 있다. 또한 관계성을 통한 다양한 정보를 획득하기 위해 추가적인 연산을 수행해야 하며, 따라서 MRA의 개발 목적이자 MRA에서 해결하고자 했던 목표인 상이한 시스템 간의 데이터 이질성을 해결하지 못한다. 따라서 이 논문에서는 MDR에 저장된 정보의 상호운용성을 위해 확장된 MRA 인터페이스를 제안한다. 이를 위해 확장된 MRA의 구현 시 적용되어야 하는 메타 모델들 간의 관계성들을 나열하고 표준 질의 언어인 SQL로 관계성을 표현하기 위한 방법을 제안한다. 또한 구현 결과 및 평가 결과를 통해 제안 방법의 장점을 보인다.

3. eMRA

이 장에서는 MRA의 문제점을 정의하고 이를 해결하기 위한 접근 방법을 기술한다. 또한 MRA의 문제점을 개선한 eMRA의 구현하기 위한 방법을 서술한다.

MRA는 다양한 MDR 시스템에 등록되어 있는 데이터에 접근하기 위한 표준 인터페이스이다. 하지만 MRA는 MDR에 정의하고 있는 개념 즉, 클래스 간 관계성을 고려하지 않고, 각각의 클래스에 해당하는 사용자 정의 타입만을 제

공한다. 즉 MDR에 정의된 클래스 간의 다양한 관계정보를 반영하지 못하므로 사용자들이 다양한 클래스로부터 데이터를 접근하고자 할 경우 추가적인 비용이 요구된다.

Fig. 2는 MDR에 정의된 클래스와 클래스들의 상속 관계가 포함된 메타모델이다. Registered_Item은 Identified_Item을 상속받으며 Administered_Item과 Attached_Item은 Registered_Item의 하위 클래스 관계다. 이처럼 MDR에는 다양한 관계 정보가 정의되어 있다. 그러나 MRA에서는 메타 모델의 개념요소, 즉 클래스를 하나의 사용자 정의 타입으로 사상하고 있을 뿐, 클래스들 간의 관계성에 대한 처리 방법이 없다. Table 1은 MRA에서 Fig. 2에 나타내고 있는 클래스들을 MRA에서 정의한 것이다. Table 1에서 나타난 것처럼 Registered_Item 클래스와 Identified_Item 클래스 간의 관계 정보가 정의 되어 있지 않기 때문에 상속관계 정보가 누락되며, 사용자가 상위 클래스의 정보를 검색하기 위해서는 추가적인 인터페이스의 구현이 요구된다.

따라서 제안하는 eMRA는 이러한 추가적인 비용이 발생하지 않도록 MDR에 존재하는 관계정보를 반영해야한다. 먼저, MDR에 존재하는 관계정보를 나열하고 그것들을 구현하는 방법을 상세히 서술한다.

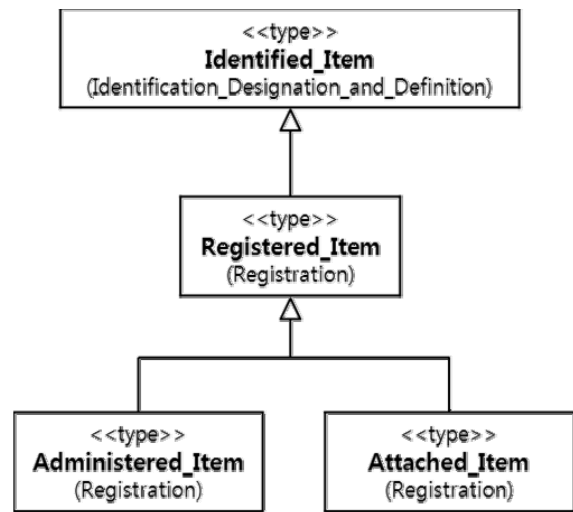


Fig. 2. Part of the MDR classes

Table 1. Specification of MDR concepts in the MRA

MRA_Identified_Item Type CREATE TYPE MRA_Identified_Item AS(...)
MRA_Registered_Item Type CREATE TYPE MRA_Registered_Item AS(...)

3.1 내포 구조 타입 (Nested structured type)

한 개의 타입에는 여러 개의 속성이 존재할 수 있으며, 각각의 속성은 자신의 데이터타입을 가진다. 데이터타입은 "Predefined type" 즉, 사전 정의된 타입과 사용자 정의 타입으로 구분된다. 사전 정의된 타입이란 데이터베이스 시스

템에 기본적으로 정의된 타입을 말하며 예로는 integer, char, date 등이 있다. 그러나 Fig. 3에 기술된 Computer 타입은 사전 정의 타입이 아닌 사용자가 정의한 타입을 가지는 속성들로 구성되어 있다. 다시 말해서, Input_Device, Output_Device가 내포 구조 타입이며, 내포 구조 타입은 다른 타입의 속성 뿐 만 아니라 테이블의 속성으로 사용 가능하며, 한 개의 사용자 타입 안에는 여러 개의 내포 타입이 정의될 수 있으며, 그 내포 타입 안에도 새로운 내포 타입이 생성될 수 있다.

Table 2. Implementation example of nested structure type

```

create type Input_Device as object (
    mouse          varchar2(10),
    keyboard       varchar2(10)
);

create type Output_Device as object (
    monitor        varchar2(10),
    printer        varchar2(10),
    speaker        varchar2(10)
);

create type Computer as object (
    input          Input_Device,
    output         Output_Device
);
    
```

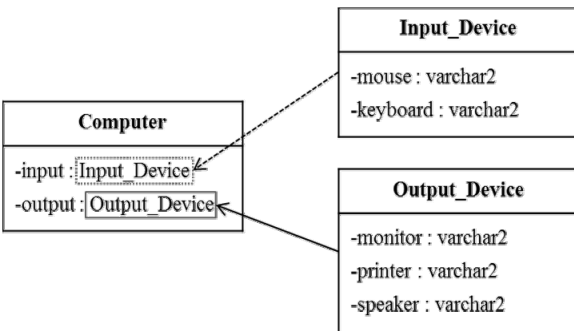


Fig. 3. Example of nested structure type relationship

Table 2는 내포 구조 타입을 생성하기 위한 문법이다. Computer 타입은 2개의 속성을 가지고 있는 것처럼 보이지만 각 속성이 사용자 정의 타입으로 정의되어 있기 때문에, Input_Device와 Output_Device 타입이 가지고 있는 속성들까지 모두 포함한다.

MDR은 Fig. 4와 같이 한 개의 클래스 안에 다른 클래스가 하나의 속성으로 정의되어 있는 관계들을 지닌다.

Fig. 4에서 Reference_Document는 MDR에서 정의한 하나의 클래스로서, 사전 정의된 타입(Text, String)들과 사용자 정의 타입(Organization, Language_Identification, Document_Type)을 가진다. Table 3은 Fig. 4에서 보인 클래스들을 데이터베이스에 구현한 것이다. Reference_Document 클래스를 사용자 정의 타입으로 구현하기 위해 Organization, Language_Identification, Document_Type, Notation 클래스들이 사전에 정의되어야 한다.

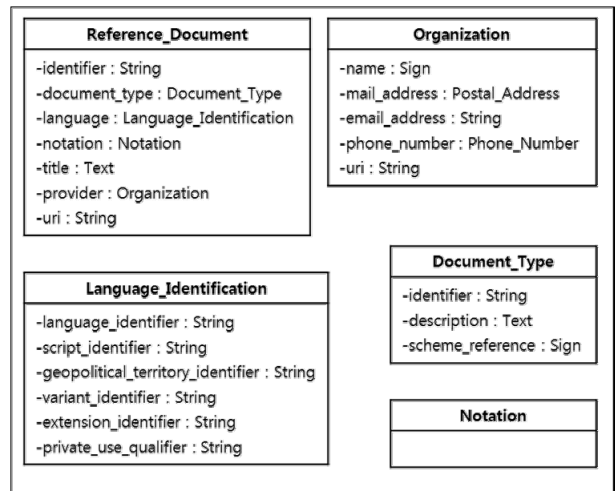


Fig. 4. Nested structure type relationships in the MDR

Table 3. Implementation of nested structure type within the MDR

```

create type Notation as object (
);

create type Document_Type as object (
    identifier      string,
    description     text,
    scheme_reference string
);

create type Organization as object (
    name           sign,
    mail_address   Postal_Address,
    email_address  string,
    phone_number   Phone_Number,
    uri            string
);

create type Language_Identification as object (
    language_identifier      string,
    script_identifier        string,
    geopolitical_territory_identifier string,
    variant_identifier       string,
    extension_identifier     string,
    private_use_qualifier    string
);

create type Reference_Document as object (
    identifier      string,
    document_type   Document_Type,
    language        Language_Identification,
    notation        Notation,
    title           text,
    provider        Organization,
    uri             string
);
    
```

3.2 관계성 처리

MDR에는 연관관계(Association), 집합연관관계(Aggregation), 복합연관관계(Composition), 상속관계(Inheritance) 등을 통해 클래스 간 관계성을 표현한다. 이 절에서는 MDR에 표현되어 있는 클래스 간의 관계성을 각각의 예제와 함께 MRA 인터페이스를 위한 해결방법에 대하여 서술한다.

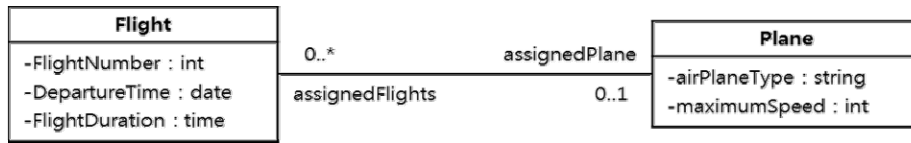


Fig. 5. Example of association relationship

1) 연관관계 처리방법

Fig. 5는 연관관계의 예를 보인다.

연관관계를 데이터베이스에 적용시키려면 기본 키 (Primary key), 외래 키(Foreign Key) 제약이 필요하다. 하지만 기본 키, 외래 키 제약은 사용자 정의 타입을 생성할 때 지정할 수 없고, 사용자 정의 타입을 기반 한 테이블인 타입형 테이블(Typed table)을 선언 할 때만 키 제약조건을 지정할 수 있다. 타입형 테이블은 사용자 정의 타입을 기반으로 실제 데이터를 물리적인 데이터베이스에 저장하기 위한 것이다. Table 4는 사용자 정의 타입과 그 사용자 정의 타입을 기반으로 한 타입형 테이블을 보인다.

Table 4. Creation example Test_table_PK

```
create type test_PK as object (
  attribute_1 integer,
  attribute_2 char,
  attribute_3 varchar2(10)
);
create table test_table_PK OF test_PK;
```

생성된 타입형 테이블은 test_PK 타입을 기반으로 한 테이블이기 때문에 test_PK 타입의 속성을 모두 포함한다. 또한 타입형 테이블인 test_table_PK를 생성할 때 기반이 되는 타입인 test_PK 타입의 속성 외에 그 어떠한 다른 속성도 추가할 수 없다. 선언된 타입형 테이블 table_PK 테이블에 기본 키, 외래 키 제약 조건을 적용하는 예제는 Table 5와 같으며, test_PK 타입에 명시되어 있는 attribute_1을 기본 키로 지정한 예를 보인다.

Table 5. Example of Applying primary key

```
create table test_table_PK of test_PK (
  primary key (attribute_1)
);
```

사용자 정의 타입 생성 시 기본 키를 지정할 수 없기 때문에 타입형 테이블을 선언할 때 기본 키를 지정한다. 다음은 Table 6에서 지정된 기본 키를 참조한 외래 키 정의 예제다.

test_table_FK 테이블은 test_FK 타입을 기반으로 한 타입형 테이블이고, 두 개의 속성 attr_1, attr_2를 가진다. Table 6은 이 두 개의 속성 중, attr_2는 test_pk의 기본 키를 참조한다. 그러므로 'foreign key(attr_2)' 구문을 이용하여 attr_2를 외래 키로 지정하고 한다. 마지막으로, 'ON

Table 6. Foreign key definition example

```
create type test_FK as object (
  attr_1 integer,
  attr_2 integer
);
create table test_table_FK of test_FK (
  foreign key (attr_2) references test_table_PK (attribute_1)
  ON DELETE SET NULL
);
```

DELETE SET NULL', 'ON DELETE SET NULL' 제약을 추가함으로써 기본 키가 삭제되었을 때, 연관관계를 맺는 테이블의 외래 키 값들을 NULL 값으로 초기화 시키는 제약사항이 추가된다. 이를 통해 두 테이블 간 관계를 삭제할 때 외래 키 값들은 NULL 값으로 변경시킨다.

Fig. 6은 앞에서 MDR에 존재하는 연관관계 예를 보인다.

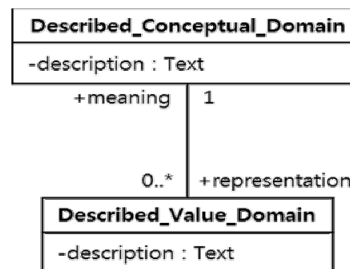


Fig. 6. Example of association relationship in the MDR

Table 7은 Fig. 6에서 보이는 연관관계를 구현을 위한 SQL 정의문을 보인다.

Fig. 6에서, Described_Conceptual_Domain 클래스는 Described_Conceptual_Domain_ID 속성을, Described_Value_Domain 클래스는 Described_Value_Domain_ID 속성을 각각 지닌다. 현재 MDR 표준에는 기본 키와 외래 키에 대한 명시가 없으므로 이 논문에서는 각각의 클래스들을 식별할 수 있는 ID 값을 생성하여 기본 키로서 정의한다. 또한 기본 키를 참조할 외래 키를 생성하여 기본 키와 외래 키 관계를 처리한다. Described_Value_Domain 클래스가 Described_Conceptual_Domain 클래스를 참조하므로 Described_Conceptual_Domain 클래스를 기반으로 한 타입형 테이블인 TAB_Described_Conceptual_Domain 테이블을 정의할 때, Described_Conceptual_Domain_ID 속성을 기본 키로 정의하고, TAB_Described_Value_Domain 테이블을 정의할 때, FK_Described_Conceptual_Domain_ID 속성을 외래 키로 지정한다.

Table 7. Implementation of association relationship

```

create type Described_Conceptual_Domain as object (
  Described_Conceptual_Domain_ID varchar2(20),
  CD_description text
);

create type Described_Value_Domain as object (
  Described_Value_Domain_ID varchar2(20),
  FK_Described_Conceptual_Domain_ID varchar2(20),
  VD_description text
);

create table TAB_Described_Conceptual_Domain of
Described_Conceptual_Domain(
  primary key (Described_Conceptual_Domain_ID)
);

create table TAB_Described_Value_Domain of
Described_Value_Domain (
  foreign key (FK_Described_Conceptual_Domain_ID)
references TAB_Described_Conceptual_Domain
(Described_Conceptual_Domain_ID) ON DELETE SET
NULL
);
    
```

2) 집합연관관계 및 복합연관관계 처리방법

집합연관관계(Aggregation)와 복합연관관계(Composition)는 연관관계의 특별한 형태로서, 먼저 집합연관관계를 데이터베이스에 표현하기 위해서는 기본 키와 외래 키를 정의하여 각 클래스의 생명주기의 독립성을 보장해야 한다. 복합연관관계를 표현하기 위해서는 'ON DELETE CASCADE' 제약을 추가한다. 'ON DELETE CASCADE' 제약은 기본 키가 삭제되면 외래 키도 삭제시키는 제약으로서, 전체 클래스와 부분 클래스의 생명주기가 같은 복합연관관계의 특징을 반영한다. Fig. 7은 MDR 메타모델에 포함되어 있는 집합연관관계 및 복합연관관계의 예로서, Table 8은 이에 대한 처리를 위한 구현 방법을 보인다.

Table 9는 Fig. 7의 Dimensionality 클래스와 Unit_Of_Measure 클래스의 집합연관관계를 구현한 것이다. 앞서 언

Table 8. Implementation of aggregation relationship

```

create type Namespace as object (
  Namespace_ID varchar2(20),
  naming_authority Organization,
  one_name_per_item_indicator boolean,
  one_item_per_name_indicator boolean,
  mandatory_naming_convention_indicator boolean,
  shorthand_prefix string,
  scheme_reference sign
);

create type Scoped_Identifier as object (
  Scoped_Identifier_ID varchar2(20),
  FK_Namespace_ID varchar2(20),
  identifier string,
  version string,
  full_expansion string,
  shorthand_expansion string
);

create table TAB_Namespace of Namespace (
  primary key (Namespace_ID)
);

create table TAB_Scoped_Identifier of Scoped_Identifier(
  foreign key (FK_Namespace_ID)
references TAB_Namespace (Namespace_ID)
ON DELETE CASCADE
);
    
```

급하였듯이, 각각의 타입을 생성할 때 관계성을 반영할 수 없다. 그러므로 타입을 기반으로 한 타입형 테이블을 각각 생성함과 동시에 기본 키와 외래 키를 지정하고 제약을 추가함으로써 관계성을 처리한다.

TAB_Unit_Of_Measure 테이블의 FK_Dimensionality_ID 속성을 외래 키로 지정한다. 이 속성은 TAB_Dimensionality 테이블의 기본 키인 Dimensionality_ID 속성에 사상한다. 오라클 11g의 외래 키 제약은 기본적으로 'ON DELETE NO ACTION' 제약이 내포되어 있다. 이는 기본 키가 삭제되어도 외래 키는 아무 영향을 받지 않는 제약으로서, ISO/IEC 9075 제2부에 정의되어 있다. 그러므로 집합연관관계는 기본 키와 외래 키 정의로써 데이터베이스에 표현이 가능하다.

3) 상속관계 처리방법

객체 지향의 가장 중요한 개념 중 하나인 상속관계는 하나의 클래스(자식 클래스)가 또 다른 클래스(슈퍼 클래스)를 상속 받았다면, 상속 받은 자식 클래스는 부모 클래스의 기능을 그대로 이용할 수 있다는 것을 의미한다. 또한 자식 클래스는 부모 클래스에서 상속받은 기능 외에 자신만의 기능을 정의하여 사용한다.

Fig. 8은 상속관계의 예다. Child 클래스는 Parent 클래스가 포함하는 속성 Parent_Attr1, Parent_Attr2, Parent_Attr3을 모두 포함할 뿐 아니라 Child 클래스 고유의 속성인 Child_Attr1, Child_Attr2도 포함한다. 상속관계를 데이터베이스에 표현할 때는 Table 10에서와 같이 'UNDER' 키워드를 사용한다.

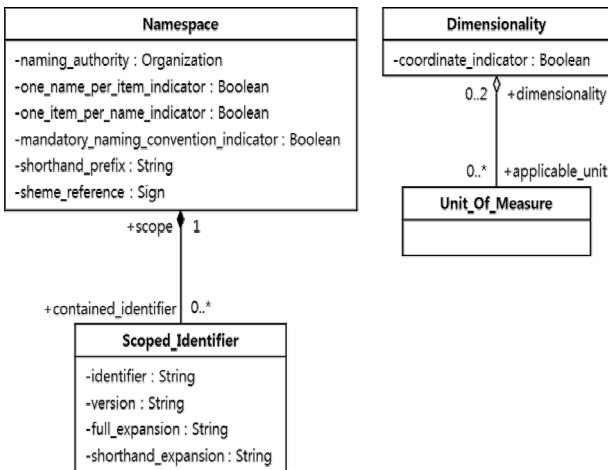


Fig. 7. Example of aggregation and composition relationship in the MDR

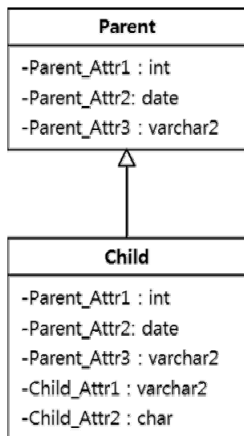


Fig. 8. Example of inheritance relationship

Table 9. Implementation of inheritance relationship

```

create type Parent as object (
  Parent_Attr1      int,
  Parent_Attr2      date,
  Parent_Attr3      varchar2
);
create type Child as object UNDER Parent(
  Child_Attr1       varchar2,
  Child_Attr2       char
);
    
```

Table 10과 같이 'UNDER' 키워드를 사용하여 상속관계를 표현하였다. 그러므로 Child 타입은 Parent 타입의 속성들을 모두 가지고 있기 때문에 Child 타입을 선언할 때 Parent 타입의 속성을 선언할 필요 없이 Child 고유의 속성만 선언한다. Fig. 9는 MDR에 있는 상속관계다.

Organization 클래스를 상속받는 Registration_Authority 클래스는 Organization 클래스가 가진 속성을 가지고, 더불어 자신만의 속성도 가진다. Table 11은 Fig. 9의 상속관계 예제에 대한 구현 결과를 보인다.

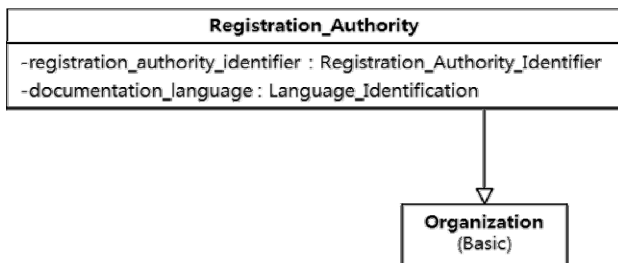


Fig. 9. Inheritance relationship in the MDR

앞서 언급한 'UNDER' 키워드로 Organization 타입을 상속받은 Registration_Authority 타입을 생성한다. 생성된 Registration_Authority 타입은 자신이 포함하는 registration_authority_identifier, documentation_language 속성 뿐 아니라 부모 타입인 Organization의 속성을 포함한다.

Table 10. Implementation of inheritance relationship

```

create type Organization as object (
  Organization_ID   varchar2(20)
);
create type Registration_Authority as object UNDER
Organization (
  Registration_Authority_ID   varchar2(20),
  FK_Organization_ID         varchar2(20),
  registration_authority_identifier   Registration_Authority_Id,
  documentation_language     Language_Identification
);
create table TAB_Organization of Organization (
  primary key (Organization_ID)
);
create table TAB_Registration_Authority of
Registration_Authority (
  foreign key (FK_Organization_ID) references
  TAB_Organization (Organization_ID)
);
    
```

상속관계를 강건하게 하기 위해서 부모 클래스와 자식 클래스에 각각 기본 키와 외래 키를 적용한다. 이를 통해 자식 클래스는 부모 클래스의 구조를 포함할 뿐 아니라 부모 클래스와의 연관성도 도출할 수 있다.

4. 실험 및 비교평가

4.1 스키마 구조

실험 평가를 위해서는 MDR 시스템을 구축해야 한다. 이 논문에서는 MDR 시스템의 프로토타입을 구현하며, 구현을 위한 DBMS로는 Oracle 11g를 이용한다. Fig. 10은 구축된 시스템의 스키마 구조를 보여 주며, 스키마는 앞서 기술한 MDR에 존재하는 관계정보(연관관계, 집합연관관계, 복합연관관계, 상속관계)를 명확하게 반영한다.

4.2 실험 및 평가 결과

이 절에서는 MRA와 eMRA의 질의처리 효율성을 비교하기 위한 실험을 수행하여 평가한다. 비교 평가 항목은 질의 모델링 시간과 질의문 필터링 개수로 구분한다.

질의 모델링 시간은 시스템에 질의하는 전체 과정을 의미한다. MRA와 eMRA 시스템에 질의를 수행할 때 관계정보의 반영이 어떤 영향을 미치는지를 비교한다. 이 논문에서는 MRA에 관계정보를 포함한 질의를 수행할 때 추가적인 연산이 요구되기 때문에 eMRA에서 질의를 수행하는 시간보다 많은 시간이 소요될 것이라고 가정한다.

질의문 필터링 개수는 시스템의 신뢰성을 측정하는 평가 요소이다. 관계정보를 반영함으로써 MRA와 eMRA에 질의를 수행했을 때 보이는 결과의 차이를 비교한다. eMRA는 MDR의 관계정보를 반영하므로 관계성을 고려한 질의를 수행할 때 누락되어야 하는 결과를 올바르게 누락시킨다고 가정한다. 그러나 MRA는 관계정보를 반영하지 않는다. 따라서 결과에서 제거되어야 하는 정보를 누락시키지 않고 도출

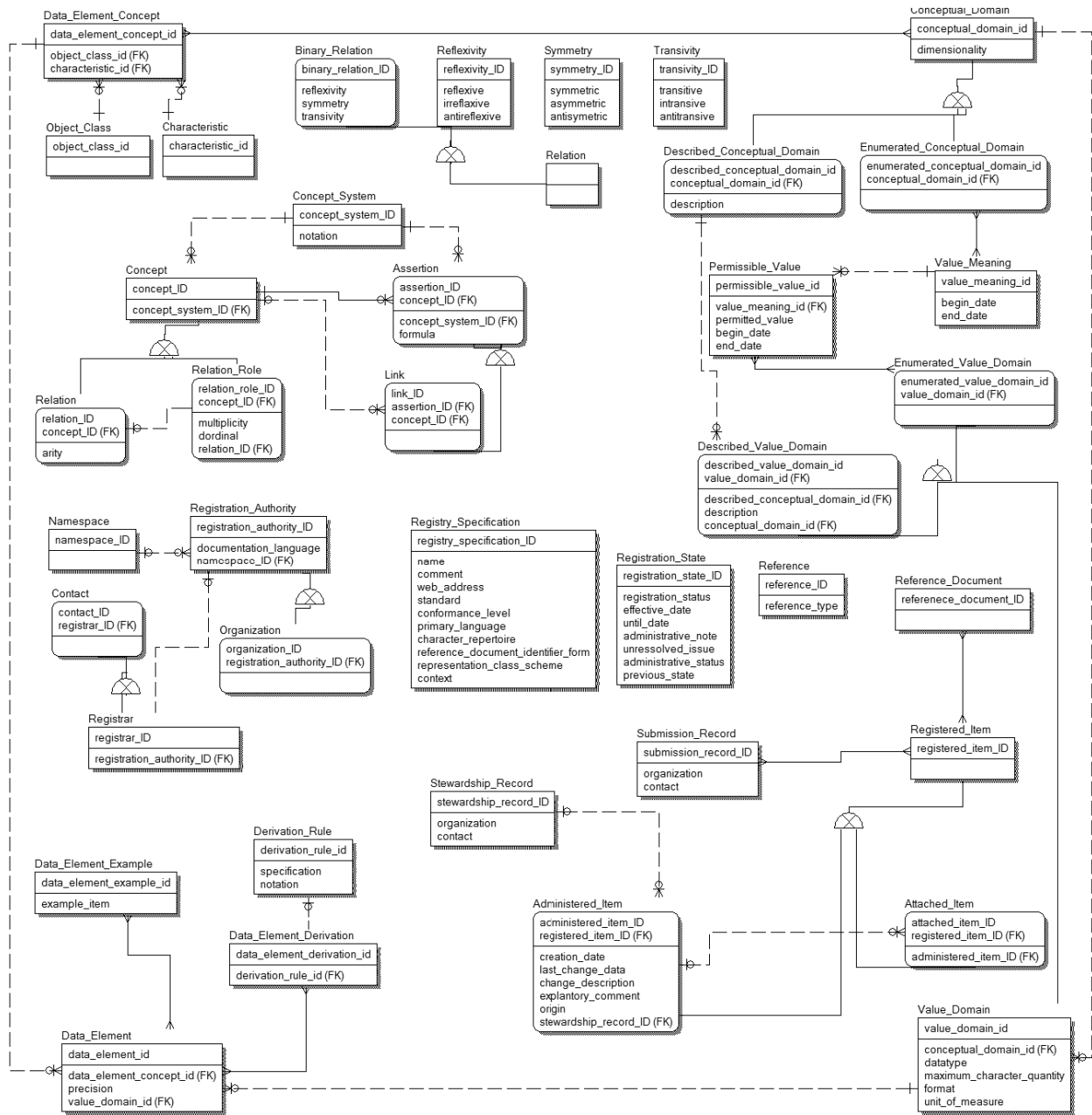


Fig. 10. The schema structure of proposed interface

한다고 가정한다. 실험을 위해 다음과 같은 전제조건을 정의한다. 먼저, MRA와 eMRA는 관계정보의 유무를 제외한 나머지 조건은 동일하다. 나머지 조건은 테이블의 이름, 수, 테이블 내 존재하는 속성의 이름과 수 그리고 저장된 데이터를 모두 포함한다. 또한 MDR 표준에 정의된 클래스는 하나의 테이블로 구축한다.

시스템에 수행하는 질의는 단순 검색질의와 복합 검색질의로 구분하여 정의한다. 복합 검색질의는 연관관계, 집합연관관계, 복합연관관계, 상속관계를 포함한다. 단순 검색질의는 단일 테이블에서 데이터를 검색하는 질의다. 반면 복합 검색질의는 두 개 이상의 테이블에서 관계성을 고려하여 데이터를 검색하는 질의다. 따라서 MRA 시스템에서 복합검색

질의를 수행시켜 정확한 결과를 얻으려 한다면 결과데이터를 정제하는 추가적인 연산이 요구된다.

1) 질의 모델링 시간

이 절에서는 3.2절에서 정의한 질의 유형을 바탕으로 실제 시스템에 질의하는 과정을 정의한다. 또한 정의한 과정을 수식화하여 MRA와 eMRA 시스템의 질의처리 효율성을 비교한다. 실험은 수식화한 질의 모델링 과정을 시뮬레이션을 통해 시간을 측정한다.

사용자는 시스템에 질의하기 위해 크게 다음 4단계의 과정을 거친다.

1. 메타데이터추출: 메타모델 정보를 분석하는 과정

2. 관계정보확인: 실제 데이터베이스시스템에 접근하여 릴레이션과 속성을 검색하고 관계정보가 정의되어있는지 확인하는 과정
3. 관계정보생성: 관계정보가 정의되어 있지 않은 경우 시스템에 관계정보를 정의하는 과정
4. 질의생성: 질의를 수행하는 과정

메타데이터추출 과정에서는 논리적인 메타모델의 정보를 분석하고 클래스의 상세한 정보도 분석 가능하다. 상세한 정보란 속성의 개수, 속성의 데이터타입, 관계정보를 포함한다. 사용자는 첫 번째 과정을 거친 후에 어떤 클래스의 어떤 속성을 검색할 것인지 결정할 수 있다. 또한 복합 검색 질의를 할 경우, 질의에 포함시킬 관계정보 개수도 결정 가능하다. 그 후 사용자는 실제 데이터베이스 시스템에 접근하여 일치하는 릴레이션과 속성을 검색해야한다.

관계정보확인 과정에서는 데이터베이스에 접근하여 릴레이션과 속성을 검색하고 메타모델에서 분석한 클래스와 속성이 일치하는지 확인한다. 또한 관계정보가 데이터베이스에 정의되어 있는지 확인한다. 만약 관계정보가 정의되어 있지 않으면 복합 검색질의를 할 수 없다. 따라서 관계정보가 정의되어 있지 않은 시스템에서 복합검색 질의를 수행하려면 추가적인 연산이 필요하다. 관계정보생성 과정에서는 앞서 수행한 관계정보가 정의되어 있지 않을 경우 수행하는 과정이다. MRA 시스템은 MDR에 명시된 관계정보를 정의하고 있지 않으므로 복합 검색질의를 수행할 수 없다. 따라서 복합 검색질의를 수행할 수 있도록 관계정보를 생성해야한다. 그러나 제안한 eMRA 시스템은 MDR에 명시된 관계정보를 명확하게 정의했기 때문에 관계정보를 확인하는 단계부터 생성하는 단계까지의 과정을 수행할 필요가 없다. 질의생성 과정에서는 실제 데이터베이스 시스템에 질의를 입력하여 결과 값을 도출하기까지의 모든 과정이다. 이 과정을 수식으로 나타내기 위해 Table 11에서는 MRA 시스템과 eMRA 시스템에 질의하는 과정을 비교하기 위한 변수 정의를 보인다.

Table 11. Variables that are used in query modeling and description

Variable	Description
tQD	Query modeling time
tMA	Required time that analyze metamodel
nR	The number of relationship between relations that included in the query
tRS	Time comparison of the class of metamodel and the relation of database.
tAS	Time comparison of the class attribute of metamodel and the relation attribute of database.
tCR	The required time to confirm whether the relationship is participate in relations is defined in database
tGR	Relationship generated time
tQC	Query execution time

Table 11에서 정의한 변수들을 이용하여 MRA 시스템에 질의하기 위한 과정을 나타내면 수식 1과 같다. 앞서 언급하였듯이 질의모델링은 메타모델 분석 과정, 실제 데이터베이스 시스템에 접근하여 릴레이션과 속성을 확인하는 과정 및 질의 수행 과정을 포함한다. MRA 시스템은 MDR의 관계정보가 반영되어있지 않기 때문에 관계정보가 정의되어 있는지 확인하는 시간이 요구된다. 또한 관계정보가 정의되어 있지 않을 경우 필요한 수만큼 관계를 정의하는 시간이 추가로 요구된다. 이 시간은 질의에 포함되는 관계정보 개수가 많아질수록 증가한다.

$$tQD = tMA + tRS + tAS + \sum_{i=1}^{nR} tCR_i + \sum_{j=1}^{nR} tGR_j + tQC \quad (1)$$

이 논문에서 제안한 eMRA는 MDR의 관계정보를 이미 반영했기 때문에 질의 모델링 시간을 단축시킨다. 수식 2는 eMRA 시스템에 질의 시간을 계산하는 수식이다. eMRA는 관계정보를 반영했기 때문에 관계정보가 정의되어 있는지 확인하고 생성하는 절차를 생략할 수 있다. 따라서 메타모델을 분석한 후, 데이터베이스 시스템에 접근하여 릴레이션과 속성을 확인하는 과정이 완료되면 바로 질의를 수행할 수 있다. eMRA 시스템의 질의 모델링에서는 관계정보 개수가 전체 모델링 시간에 영향을 주지 않는다. 따라서 전체 질의 모델링 시간이 단축된다.

$$tQD = tMA + tRS + tAS + tQC \quad (2)$$

Fig. 11은 관계정보 개수에 따른 질의 모델링 시간을 비교한 그래프이다. 질의모델링 시간은 과정에 개입하는 사람의 개인차에 의존적이기 때문에 수식에 사용되는 변수는 0부터 1사이의 랜덤 값으로 실험한다. 또한 충분한 시뮬레이션을 통해 보다 신뢰성 있는 결과를 내기 위해 총 100번의 실험을 거쳐 평균값을 계산한다.

eMRA는 MDR의 관계정보를 반영했기 때문에 관계정보가 정의되어 있는지 확인하고 생성하는 시간 $(\sum_{j=1}^{nR} tGR_j, \sum_{i=1}^{nR} tCR_i)$ 이 생략된다. 따라서 질의에 포함된 관계정보의 개수 nR 의 개수가 증가해도 전체 질의 모델링 시간은 증가하지 않는다. 반면 MRA는 nR 의 개수가 증가할수록 $\sum_{j=1}^{nR} tGR_j$ 과 $\sum_{i=1}^{nR} tCR_i$ 이 증가하게 되므로 질의 모델링 시간이 증가한다. 실험 결과, 관계정보의 개수가 많아질수록 큰 차이가 나는 것을 확인하였고 eMRA는 관계정보개수에 영향 받지 않는 것을 확인하였다.

2) 질의문 필터링 개수

이 절에서는 삽입, 삭제, 갱신에 관한 질의를 통해 eMRA의 질의 처리 결과에 대한 정확성을 평가한다. eMRA의 참조무결성 여부를 판단하기 위해 삽입, 삭제, 갱신 질의를 수행한 결과를 평가요소로 정의하였다. MRA 시스템은 MDR

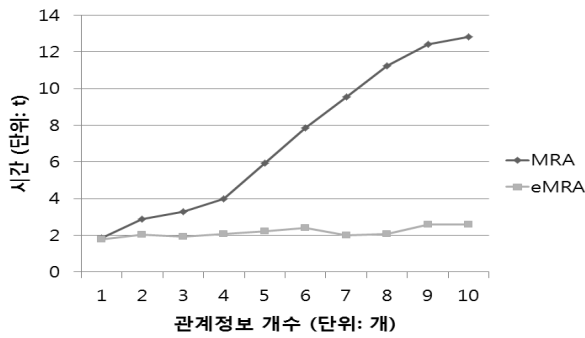


Fig. 11. Time comparison of query modeling in MRA and eMRA

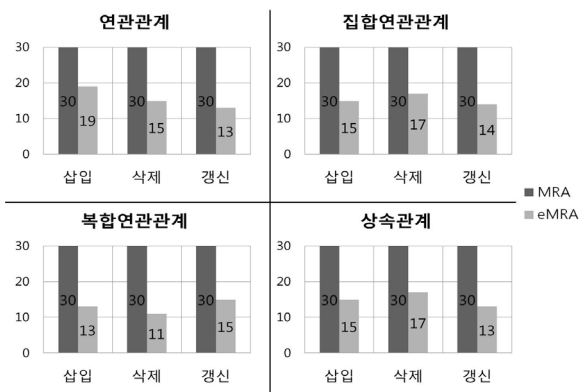


Fig. 12. Query results of each relationship

의 관계정보를 반영하지 않으므로 질의 결과에서 누락되어야 하는 정보들이 누락되지 않는다. 이는 시스템의 신뢰도를 하락시키는 요인이 된다.

Fig. 12는 MRA와 eMRA를 기반으로 한 시스템에 삽입, 삭제, 갱신에 관한 질의를 각각 30개씩 샘플링하여 시스템에 질의한 후, 질의가 실제 수행된 횟수를 비교하였다. 각 질의는 관계성을 반영한 단일 테이블에 실행하는 질의이다. Table 12은 각 질의에 대한 상세설명이다.

Table 12. Insert, delete, update query

Query	Description	
insert	type	Query that add a new tuple in table
	query	Primary key must be unique in table, so if added tuple has a duplicate primary key, system should stop the execution
delete	type	Query to remove the existing tuple in the table
	query	If there is a foreign key that references the primary key in another table, system should stop the execution
update	type	Query to change existing tuple in the table
	query	The system should stop the execution if changed primary key is duplicated, or if there is a foreign key that reference the primary key in another table

eMRA를 적용한 시스템은 참조 무결성 제약조건과 관계 정보를 반영하였기 때문에, 제약 조건을 위배하는 질의가 수행될 수 없다. 이 논문에서는 이러한 특성을 질의 필터율이라 정의한다. 즉, 질의 필터율이란 관계성에 어긋나는 질의가 수행되었을 때 시스템에서 얼마나 질의를 걸러낼 수 있는지에 대한 특성을 의미한다. Fig. 13에서 MRA의 질의 필터율은 0%인 것에 반해, eMRA에서는 평균 33.75%의 필터율을 보였다. eMRA에서는 참조 무결성 제약 조건이 반영되었기 때문에 데이터의 삽입 삭제 갱신의 질의들이 필터링됨을 알 수 있다. 기본적으로 클래스들 간의 상속관계는 다른 관계와는 달리 기본 키 제약을 적용하지 않아도 관계를 정의할 수 있기 때문에 삭제, 갱신 질의에 문제가 없다. 그러나 상속받는 자식 테이블은 부모 테이블의 속성도 포함하고 있다. 따라서 데이터를 삽입하는 경우, 자식 테이블의 속성과 부모 테이블의 속성 모두를 반영하여 데이터를 삽입하지 않으면 에러가 발생한다. eMRA는 이렇게 에러가 발생하는 질의를 필터링함으로써, 시스템 전체의 참조무결성을 보장하고 표준 인터페이스의 역할을 완전하게 수행한다.

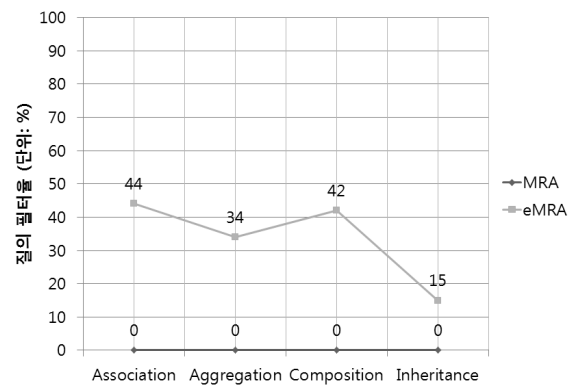


Fig. 13. Query filter rate

5. 결론 및 향후 연구

ISO/IEC 11179 MDR 표준은 메타데이터를 등록하고 인증하여 표준화된 데이터를 유지 및 관리하기 위해 제정되었다. 또한 표준화된 데이터의 관리를 통해 상호운용성을 향상시킬 수 있다. 이러한 장점을 통해 의료, 서지, 환경, 국방 등 다양한 분야에서 MDR을 기반으로 한 시스템을 개발하고 운영하고 있다. 그러나 MDR 표준은 논리적인 모델만 명시할 뿐 구체적인 물리적 구현모형을 제공하고 있지 않다. 따라서 개발된 시스템들이 MDR 표준을 기반으로 하고 있음에도 불구하고 서로 다른 구조와 접근방법을 가진다. 그 결과 다양한 MDR에 대한 일관된 접근을 어렵게 하고 재사용성 및 활용성 저하를 야기한다. 이러한 문제점을 해결하기 위해 ISO/IEC 13249-8 MRA가 제안되었다.

MRA는 MDR을 기반으로 한 시스템에 접근할 수 있는 표준 인터페이스를 제안한다. 그러나 MRA는 MDR에 접근하기 위한 기본적인 데이터타입만을 제공한다. MDR 표준에

는 클래스와 클래스 사이에 존재하는 관계정보가 존재한다. 이러한 관계정보를 시스템이 반영하지 않는다면 질의처리 효율성이 저하된다. 따라서 질의처리의 효율성을 높이기 위해 추가적인 연산이 요구된다.

따라서 이 논문에서는 기존 연구의 문제점을 극복할 수 있는 확장된 인터페이스 eMRA를 제안하였다. MDR 표준에 존재하는 4가지 관계정보(연관관계, 집합연관관계, 복합연관관계, 상속관계)를 시스템에 적용할 수 있도록 구현하였다. 제안한 eMRA의 질의처리 효율성을 평가하기 위해 관계정보를 반영하지 않은 MRA와 비교하였다. 평가항목으로서 질의 모델링 시간과 질의문 필터링 개수를 정의하였다. eMRA는 MDR의 관계정보를 반영함으로써 관계정보 확인 단계와 관계정보 생성 시간을 생략함으로써 질의 모델링의 효율성을 보였다. 또한 관계정보 반영으로 인해 시스템의 참조무결성이 보장되었고, 이를 통해 복합 검색질의를 수행하였을 때 MRA에 비해 높은 정확성을 보였다.

이 논문은 기존 MRA가 가지고 있던 문제점인 관계정보를 적용하지 않았다는 점을 개선하여 표준 인터페이스로서의 가치를 높였다. 이를 통해 서론 부분에서 소개한 다양한 방면에서 사용되고 연구되고 있는 MDR 시스템에 적용 가능하다. 각 시스템은 MDR을 기반으로 구축되었기 때문에 제안한 eMRA를 적용할 수 있다.

이 논문에서는 오라클을 이용한 구현 모델을 보였다. 향후에는 보다 다양한 데이터베이스 모델 및 데이터베이스 관리시스템을 이용한 구현 모델을 개발하고 각각의 특징에 대한 연구가 요구된다.

참 고 문 헌

- [1] R. K. Pon, D. J. Buttler, "Metadata Registry, ISO/IEC 11179," Encyclopedia of Database Systems, pp.1724-1727, 2009.
- [2] Y. Park, J. Kim, "Metadata registry and management system based on ISO 11179 for cancer clinical trials information system," Proc. of the AMIA Annual Symposium Proceedings, Vol.2006, pp.1056, 2006.
- [3] J. Davies, S. Harris, Ch. Crichton, A. Shukla, J. Gibbons, "Metadata standards for semantic interoperability in e-government," Proc. of the 2nd International Conference on Theory and Practice of Electronic Governance, pp.67-75, 2008.
- [4] J. Ostausberg, M. Löbe, P. Verplancke, J. Drepper, H. Herre, M.Löffler, "Foundations of a Metadata Repository for Databases of Registers and Trials," Proc. of European Federation for Medical Informatics, pp.409-413, 2009.
- [5] L. Bountouri, C. Papatheodorou, V. Soulikias, M. Stratis, "Metadata interoperability in public sector information," Journal of Information Science, Vol.35, No.2, pp.204-231, 2008.
- [6] D. Shin, E. Jeong, D. Jeong, S. Park, D. Baik, "A Model for Metadata Registry Management Systems," Proc. of Korea Society for Simulation Fall Conference, pp.173-177, 2003.
- [7] D. Shin, D. Jeong, Y. Kim, S. Park, D. Baik, "A Query Language for Consistent Access of Metadata Registries," Journal of Korea Information Science Society: Databases, Vol.2, No.1, pp.609-623, Dec., 2004.
- [8] Australian Institute of Health and Welfare, Metadata Online Registry (METeOR), <http://meteor.aihw.gov.au/> (accessed September 2012).
- [9] US Health organizations (multiple), US Health Information Knowledgebase (USHIK), <http://ushik.ahrq.gov> (accessed September 2012).
- [10] US Environmental Protection Agency, Environmental Data Registry, <http://www.epa.gov/edr/> (accessed September 2012).
- [11] US National Cancer Institute, Cancer Data Standards Repository (caDSR), <http://ncicb.nci.nih.gov/NCICB/> (accessed September 2012).
- [12] U.S. Department of Homeland Security (DHS) and U.S. Department of Justice (DOJ), US National Information Exchange Model (NIEM), <http://www.niem.gov/> (accessed September 2012).
- [13] US Department of Justice, Global Justice XML Data Model (GJXDM), <http://www.it.ojp.gov/> (accessed September 2012).
- [14] The Library of Congress(LC), <http://www.loc.gov/> (accessed September 2012).
- [15] Canadian Institute for Health Information, CIHI Data Dictionary, <http://secure.cihi.ca/cihiweb/splash.html> (accessed September 2012).
- [16] UK Cancer Grid, Cancer Grid Metadata Registry, <http://www.cancergrid.org/> (accessed September 2012).
- [17] UK Office for Library Networking(UKOLN), <http://www.ukoln.ac.uk/> (accessed September 2012).
- [18] D. Baik, Y. Choi, S. Park, J. Lee and D. Jeong, "A 3-Layered Information Integration System based on MDRs End Ontology," The KIPS transactions: Part D, Vol.10, No.2, pp.247-260, 2003.
- [19] J. Kim, H. Park, C. Moon and D. Baik, "Design and Implementation of A Distributed Information Integration System based on Metadata Registry," The KIPS transactions: Part D, Vol.10, No.2, pp.233-246, 2003.
- [20] J. Kim, D. Jeong, J. Kim and D. Baik, "Implementation of the Metadata Registry-based Framework for Semantic Interoperability of Application in Ubiquitous Environment," Journal of Korea Society for Simulation, Vol.16, No.1, pp.11-19, 2007.
- [21] D. Jeong, H. Jeong, J. Kim, "A Study on Standardization of Query Processing of Metadata Registries," Proc. of the 2010 The Korea Society of Computer and Information Winter Conference, Vol.18, No.2, pp.65-68, 2010.
- [22] J. Melton, A. Eisenberg, "SQL Multimedia and Application Packages (SQL/MM)," Proc. of Association of Computing Machinery SIGMOD Record, Vol.20, No.4, pp.97-102, 2001.

[23] ISO/IEC 20944 Information technology - Metadata Registries Interoperability and Bindings (MDRIB), <http://jtc1s32.org> (accessed September 2012).

[24] P. Jyotichman, P. Lee, G.C. Christopher, B. Olivier, "Comparing and evaluating terminology services application programming interfaces: RxNav, UMLSKS and LexBIG," Journal of the American Medical Informatics Association (JAMIA), Vol.17, No.6, pp.714-719, 2010.

[25] ISO/IEC 9075 Information technology - Database languages - SQL - Part 2: Foundation (SQL/Foundation), <http://jtc1s32.org> (accessed September 2012).

[26] ISO/IEC 13249 Information technology - Database languages - SQL Multimedia and Application Packages - Part 8: Metadata Registry Access, <http://jtc1s32.org> (accessed September 2012).

[27] D. Jeong, H. Jeong, J. Kim, K. Jeon, S. Shin, "Development of the SQL/MM Standard for Metadata Registries," Journal of the Korea Society of Computer and Information, Vol.15, No.9, pp.9-18, Sep., 2010.

[28] Y. Joo, S. Lee, J. Kim, D. Jeong, D. Baik, "An Implementation of ISO/IEC 13249-8 with Oracle," Proc. of the Korea Computer Congress 2011, Vol.38, No.1(A), pp.120-123, 2011.

2000년~2001년 (주)지구넷 부설 연구소 선임연구원
 2002년~2005년 라임디미디어 테크놀로지 부설 연구소 연구원
 2004년~2005년 고려대학교 정보통신기술연구소 연구조교수
 2005년 Pennsylvania State University PostDoc
 2002년~2004년 TTA 표준화위원회-데이터연구회(SG08.02) 특별위원
 2004년~현 재 TTA 표준화위원회-메타데이터 표준화 프로젝트 그룹(PG406) 위원
 2005년~현 재 군산대학교 정보통계학과 교수
 2006년~현 재 데이터관리서비스 전문위원회(ISO/IEC JTC 1/SC 32 Mirror Committee) 위원
 2008년~현 재 지리정보 전문위원회(ISO/TC 211 Mirror Committee) 위원
 2009년~현 재 TTA 표준화위원회-NGIS 프로젝트그룹(PG409) 위원
 2010년~현 재 인터넷윤리실현협회 이사
 2010년~현 재 ICDL Korea 교수위원
 2010년~현 재 전북지역 과학기술정보협의회 위원
 2010년~현 재 한국과학기술정보연구원 자문위원
 2010년~현 재 한국컴퓨터교육학회 이사
 관심분야: 데이터베이스, 시맨틱 웹, 시맨틱 GIS, 유비쿼터스 컴퓨팅, 시맨틱 모바일 서비스, 클라우드 컴퓨팅



주 영 민

e-mail : wndudals77@korea.ac.kr
 2011년 고려대학교 컴퓨터정보학과(학사)
 2010년~현 재 고려대학교 컴퓨터·전파 통신공학과 석사과정
 관심분야: 데이터베이스, 메타데이터 레지스트리, 표준 인터페이스



백 두 권

e-mail : baikdk@korea.ac.kr
 1974년 고려대학교 수학과(학사)
 1977년 고려대학교 산업공학과(석사)
 1983년 Wayne State Univ. 전산학과(석사)
 1985년 Wayne State Univ. 전산학과(박사)

1986년~현 재 고려대학교 컴퓨터·전파통신공학과 교수
 1991년~현 재 (사)한국시물레이션학회 (이사/부회장/감사/회장/고문)
 1991년~현 재 ISO/IEC JTC1/SC32 전문위원회(위원장)
 1999년~2000년 고려대학교 컴퓨터과학기술연구소(소장)
 1999년~1999년 한국 DB 진흥센터 (표준연구위원)
 2000년~2003년 소프트웨어 컴포넌트 표준화 포럼(부의장)
 2001년~현 재 (사)도산아카데미(원장)
 2002년~2004년 고려대학교 정보통신대학(초대학장)
 2004년~2005년 (사)정보처리학회(부회장)
 2004년~현 재 한국 프로젝트 관리 연구회(회장)
 2009년~2010년 고려대학교 정보통신대학 학장
 관심분야: 메타데이터, 소프트웨어공학, 데이터공학, 컴포넌트 기반 시스템, 메타데이터 레지스트리, 프로젝트 매니지먼트 등



김 장 원

e-mail : ikaros1223@korea.ac.kr
 2005년 상명대학교 소프트웨어공학과(학사)
 2005년 한국과학기술연구원(KIST) 위촉연구원
 2008년 고려대학교 컴퓨터학과(석사)
 2012년 고려대학교 컴퓨터학과(박사)

관심분야: 온톨로지, 시맨틱 웹, GIS, 데이터베이스, 메타데이터 등



정 동 원

e-mail : djeong@korea.ac.kr
 1997년 군산대학교 컴퓨터학과(이학사)
 1999년 충북대학교 전산학과(이학석사)
 2004년 고려대학교 컴퓨터학과(이학박사)
 1998년 전자통신연구원 위촉연구원
 1999년~2000년 ICU부설 한국정보통신 교육원 GIS 분원 전임강사