RBAC기반의 메타데이타 레지스트리 접근제어 모델

황선홍 [O1], 김진형 [1], 정동원 [2], 김희석 [3], 백두권 [1]
[1] 고려대학교 컴퓨터 · 전파통신공학과, [2] 군산대학교 정보통계학과, [3] 삼성전자
{diebroke [O1], koolmania[1], baikdk[1] }@korea.ac.kr, djeong[2]@kunsan.ac.kr,
seogkim[3]@samsung.com

Metadata Registry Access Control Model based on RBAC

Sunhong Hwang [O1], Jinhyung Kim [1], Dongwon Jeong[2], Heuiseog Kim[3], Doo-Kwon Baik[1]
[1]Dept. of Computer Science and Engineering, Korea University
[2]Dept. of Informatics & Statistics, Kunsan National University
[3]SAMSUNG ELECTRONICS CO., LTD.

요 약

다양한 분야에서 ISO/IEC 11179를 기반으로 MDR(Metadata Registry)시스템들이 개발되었다. 그러나 현재 구축된 메타데이타 관리 시스템들은 표준을 따라서 생성되지 않아 메타데이타 간 불일치가 발생하는 문제가 있다. 그리고 메타데이타를 공유하고 교환할 수 있는 표준화된 접근방법을 제공하지 않아 MDR 시스템마다 상이한 방법을 이용하여 개발되는 문제점을 야기한다. 이러한 문제점들을 해결하기 위해 SQL/MDR이 제안되었다. SQL/MDR은 MDR에 대한 사용하기 쉬운 표준 인터페이스를 제공함으로써 반복적인 메타데이터 레지스트리 접근연산 개발 시 메타데이터 레지스트리 간 데이터 불일치를 개선할 수 있게 한다. 그러나 SQL/MDR은 검색을 위한 연산만을 지원할 뿐, MDR 구축 시 접근제어를 위한 연산은 제공하지 않아 정확하고 표준화된 MDR 구축 및 안전한 접근제어를 보장하지 못한다. 이 논문에서는 앞서 언급한 SQL/MDR문제점 중에서 안전한 접근제어를 보장할 수 있는 방법으로 MCL(Metadata Control Language)을 제안한다. MCL은 ISO/IEC 11179 Part 6에서 제안하는 사용자 그룹의 역할과 권한을 미리 정의하여 사용자를 사용자 그룹으로 할당하는 간단한 연산자를 사용함으로써 사용상의 편이성과 보안성을 증대시킨다. 또한 시스템 관리자가 쉽고 정확하게 MDR에 대한 접근제어 규칙을 쉽게 정의할 수 있게 하여 시스템 관리 시간 및 비용을 감소시킨다.

키워드: 접근제어 모델, 보안, RBAC, Metadata Registry

1. 서 론

ISO/IEC 11179 는 메타데이터 간 불일치에 의한 상호운용성 문제 해결과 정보공유 및 교환의 용이성을 위하여 개발되었다[1]. 그러나 MDR 시스템 구현 시 연산패턴을 서로 다른 방법으로 개발함으로써 시간과 비용의 낭비를 초래하고 제약조건이 고려되지 않고 개발되어 MDR 간의 이질성 문제를 발생시키는데 이러한 문제점 해결을 위해 SQL/MDR 을 제안하여 현재 표준화 진행 중이다[2,3]. 그러나 SQL/MDR 에서도 MDR 접근시스템에 대해서는 고려하지 않고 있어 보안성과 사용상의 편이성이 떨어진다. 이러한 부분에 대한 일관된 사용자 접근부분에 대한 제안이 필요하다.

MDR 접근방법은 DCL 기반으로 하는 방법이 있고, 우리가 제안하고자 하는 방법인 MCL(Metadata Control Language)을 사용하는 방법이 있다. ISO/IEC 11179 에서는 MDR 의 개발 및 관리를 위해 데이터요소에 대한 검색, 등록, 투표, 등록상태변경, 개정 및 버전관리 등과 같은 다양한 기능에 대하여 명세하고 있다.

이 연구에 참여한 연구자는 '2단계 BK21 사업'의 지원을 받았음.

ISO/IEC 11179 표준에 따라 개발된 MDR 시스템은 크게 두 가지 특징을 갖는다. 첫째, MDR 접근 시 일정한 형태의 질의 연산 패턴을 갖는다. 다시 말해, 표준에서 요구하는 필수 기능을 제공해야 하므로 모든 MDR 시스템들은 일정한 질의 패턴을 지니게 된다. 둘째, MDR 시스템은 필수적인 구성요소를 제공한다. ISO/IEC 11179 를 준수하여 시스템 개발 시 명세에서 요구하는 필수 테이블과 필수 속성이 있다.

현재까지 구축된 MDR 들은 MDR 시스템 접근 시 사용되는 연산 패턴들에 대한 일관성 있는 접근방법이 연구되지 않았다. 이는 MDR 시스템 개발 시 ISO/IEC 11179 를 준수하기 위해 매번 동일한 연산 패턴을 다른 방법으로 개발함으로써 시간과 비용의 낭비를 초래한다. 또한 표준접근 방법을 제공하지 않기 때문에 MDR 접근 시 반드시 요구되는 제약조건이 고려되지 않는 경우가 발생한다. 이는 MDR 간의 이질성 문제를 야기하고 표준화된 MDR 구축을 어렵게 한다. 그리고 ISO/IEC 11179 에서는 기능에 대한 개략적인 설명만 있을 뿐 구체적인 스키마를 제공하지 않는다. 따라서 개발된 MDR 시스템마다 다른 구조와 명칭을 갖는 스키마를 사용하게 되며

이는 결국 MDR 간의 이질성을 유발한다. 이러한 문제점 해결을 위해 ISO/IEC 11179 에서 요구하고 있는 테이블과 속성을 정의하고 동일한 패턴을 지니는 접근연산들을 분석하고 일관성 있는 접근을 위한 MDR 질의 언어를 정의 및 설계하기 위해 국제표준 질의어인 SQL[4]을 기반으로 확장된 메타데이타 레지스트리 질의 언어를 SQL/MDR 이라고 정의함으로써 MDR 시스템개발을 보다 용이하게 하였다. MDR 간의 메타데이타 공유 및 교환을 위하여 SQL 기반의 DCL 을 사용할 경우 구현하는 과정이 복잡하여 사용성이 떨어지고, 사용자 권한 부여시 복잡한 과정을 거치다 보면 접근 보안상의 오류가 발생할 가능성이 크다.

이 논문에서는 SQL/MDR 중 MDR 접근제어에 대한 문제점을 해결하여 MDR 시스템 개발과 사용성 및 보안성을 높이는데 있고 MDR 접근제어를 다루는 질의 언어를 MCL(Metadata Control Language)로 정의하였다. 구현된 MDR 접근제어 시스템은 ISO/IEC 11179 의 규칙을 준수하고 활용을 위한 필수적인 기능들을 포함하고 있어 시스템 개발 프로세스 및 MDR 접근제어 구축을 위한 지침서로서 이용될 수 있다. 또한 컴포넌트를 기반으로 설계 및 구현되었기 때문에 다양한 분야의 메타데이타 레지스트리 관리 시스템 개발을 위한 재사용이 용이하며 시스템 개발 시간과 비용을 감소시키고 일관된 정형화된 접근방법을 제공함으로써 사용의 편이성과 보안성을 향상시킨다.

이 논문의 구성은 다음과 같다. 제 2 장에서는 관련연구 부분인 MDR 개념과 기존 관리 시스템의 특징에 대하여 기술한다. 제 3 장에서는 MCL 개념 및 정의에 대하여 기술하고, 제 4 장에서는 구현 및 평가에 대하여 기술한다. 제 5 장에서는 결론에 대하여 기술한다.

2. 관련연구

2.1 ISO/IEC 11179 소개

ISO/IEC 11179 는 데이타를 쉽게 이해하고 상호운영성을 높이기 위해 국제표준 기구인 ISO/IEC JTC 1 에 의해 개발된 표준이다. 이는 상호간 데이터 공유 및 교환을 극대화하여 일관성 있는 데이타를 표현하고 구축하기 위한 국제 표준이다. MDR 은 데이터 요소들의 집합이고, 데이터 요소는 정의, 식별, 표현 및 허용 가능한 값들의 속성 집합으로 데이터를 명세하기 위한 최소 단위이다. 국내에서 구축된 MDR 은 한국전자통신연구원의 컴포넌트의 원활한 사용을 위해 구축한 컴포넌트 메타데이타 레지스트리[5], 한국과학기술정보 연구원의 서지정보 메타데이타 레지스트리[6]가 있다. 해외에서 구축된 메타데이타 레지스트리는 미국 환경청의 환경청의 환경 정보를 위한 EDR(Environmental Data Registry)[7,8], 호주 건강복지기간의 NHIK(Australian National Health Information Knowledgebase)[9,10], 미국 교통부의 ITS(Intelligent Transportation System)[11,12] 등이 있다. 이들은 모두 국제 표준인 ISO/IEC 11179 를 기반으로 구축되었으나 일관성과 표준화가 결여된 MDR 접근 방법을 사용하였고 MDR 접근 시 반드시 요구되는 제약조건을 고려하지 않아 MDR 간 이질성을 야기함으로써 국제표준인 ISO/IEC 11179 의 혼란을 발생시킨다. 이는 결국 MDR 간 메타데이타의 불일치를 야기하게 된다.

2.2 SQL/MDR

앞서 언급한 MDR 의 문제점인 표준화된 MDR 을 제공하지 않는 것을 해결하기 위해 SQL/MDR 이 개발되었다. 즉, SQL/MDR 의 목적은 메타데이타 간의 불일치를 해결하기 위해 메타데이타 레지스트리에 대한 표준 인터페이스를 제공하여 일관성 있는 접근 방법을 제공하는데 있다. 이는 MDR 시스템 구현 시 연산패턴을 다른 방법으로 개발함으로써 시간과 비용의 낭비를 초래하고 제약조건이 고려되지 않고 개발되어 발생하는 MDR 간의 이질성 문제를 해결하기 위해 SQL/MDR 을 제안하여 현재 표준화가 진행 중이다. 이미 SQL/MDR 과 유사한 접근방법들이 많은 데이터베이스 분야에서 이루어져 왔다. 공간 데이타의 표준 접근을 위한 공간 질의어(Spatial Query Language) 연구[13,14], 시간 데이터를 위한 시간 질의어(Temporal Query Language) 연구[14,15], 멀티미디어 데이타를 위한 SQL/MM[16]등이 그 예이며, 이들은 모두 국제 표준 질의 언어인 SQL 을 기반으로 확장된 질의 언어들이다.

이와 같은 연구들은 특수한 형태의 데이타들에 대해 각각 다른 구조를 지니는 데이타베이스를 중립적으로 액세스하기 위한 것이다. 즉 표준화된 액세스 방법으로 다양한 데이타베이스를 일관성 있게 접근할 수 있게 해준다. 따라서 각 구조에 독립적이고 일관성 있는 질의 모델링이 가능하며 질의 모델링 비용 및 분산된 다양한 데이터베이스로부터의 분산 처리 비용 등 효율성을 제공한다.

그러나 비록 SQL/MDR 이 일관성 있는 접근방법을 제공한다 해도 단순한 검색을 위한 방법만을 제공하며 이는 MDR 에 대한 수정 연산, 구조 정의는 물론 접근제어를 위한 추가적인 오버헤드를 요구하게 된다.

3. MCL 개념 및 정의

이 장에서는 ISO/IEC 11179 의 명세를 바탕으로 MDR 접근에 대한 인터페이스로 MCL 을 정의하고, MCL 세부 연산자인 MGRANT 와 MREVOKE 에 대하여 정의한다.

3.1 MCL(Metadata Control Language) 개념

ISO/IEC 11179 Part6 에는 사용자 그룹 및 각 사용자 그룹의 Role 이 정의되어 있다. 이러한 정보를 기준으로 하여 개별 사용자에게 정확하고 편리하게 부여할 수 있도록 하는 언어를 제안하고 이를 MCL 이라고 명명한다. 이러한 MCL 은 SQL/MDR 이 가지고 있는 사용성과 보

안성 부분의 취약점을 개선한다. MCL 은 MGRANT(GRANT FOR MDR)와 MREVOKE (REVOKE FOR MDR)가 있다.

MDR 에서 사용자 그룹에 대한 역할과 권한에 대하여 정의하고 있으므로, 기 정의되어 있는 사용자 그룹과 권한을 DBMS 상에서 구현하여 MCL 연산자로 사용자가 사용자 그룹의 권한을 주는 시스템을 구현한다. 구현된 MDR 접근제어 시스템은 ISO/IEC 11179 의 규칙을 준수하고 활용을 위한 필수적인 기능들을 포함하고 있어 시스템 개발 프로세스 및 MDR 접근제어 구축을 위한 지침서로서 이용될 수 있다. 또한 컴포넌트를 기반으로 설계 및 구현되었기 때문에 다양한 분야의 메타데이타 레지스트리 관리 시스템 개발을 위한 재사용이 용이하며 시스템 개발 시간과 비용을 감소시키고 일관된 정형화된 접근방법을 제공함으로써 사용의 편이성과 보안성을 향상시킨다. DCL 을 이용한 MDR 보안모델의 구현도 가능하지만 DCL 기반으로 구현하기 위해서는 MDR 에 대한 이해에 많은 시간이 들어가고, 구축 시에도 MDR 표준을 명확히 구현해야 하는 문제점이 있다.

3.2 MDR 사용자 그룹 Role 정의

ISO/IEC 11179 Part 6 '데이터 요소의 등록'에서는 사용자, 제출자, 책임자, 등록자, 통제위원회, 집행위원회의 6 개의 사용자 그룹으로 분류된다. 일반 사용자(Read- only user)는 메타데이타 내용을 검색(검토)하는 역할을 하고, 제출자(Submitter)는 새로운 메타데이타 관리항목을 제출하는 역할을 하고, 전문가(Stewards)는 메타데이타 관리항목의 품질을 검토 및 보장하고, 등록자(registrar)는 데이터 요소를 등록하고 관리하고 유지하는 역할을 하고, 통제위원회는 기술적 문제의 해결방향을 제시하고, 집행위원회는 MDR 전체정책과 방향에 대한 역할을 한다. 메타데이터 등록 프로세스와 관련된 등록 활동주체(RAB, Registration Acting Body)의 구성은 그림 1 과 같다.

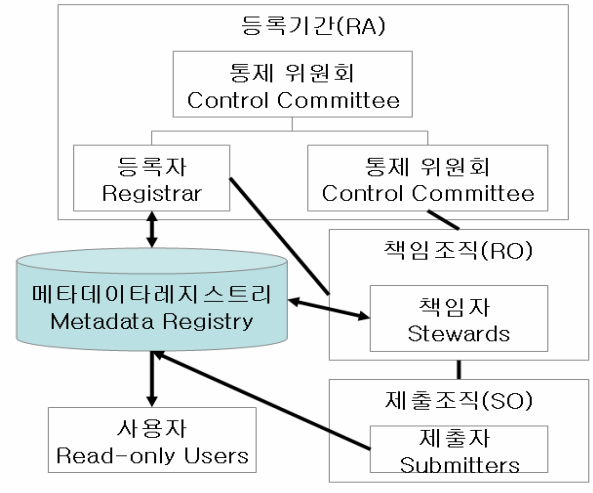

그림 1. MDR에 대한 조직별 Role과 관계

사용자 그룹별 세부 역할과 권한은 표 1 과 같다.

표 1. 사용자 그룹별 역할과 권한

User Group	Role	Privilege
Registrar	Monitoring and managing the Metadata Registry contents Proposing procedures and standard formats for the MDR Recording current registration status for Administered Items Ensuring access for authorized users Assisting in the progression of Administered Items Enforcing data registration procedures Adding new users or organizational entities	Alter, Insert, Delete, Select
Executive Committee	Establishing overall Metadata Registry policies Resolution of all business management issues Ensuring the long-term success and performance of the Metadata Registry Establishing and updating the Metadata Registry charter and strategic plans Meeting periodically in face-to-face	Alter, Insert, Delete, Select
Control Committee	Overall conduct of registration operations. Promoting the reuse and sharing of data in the metadata register Progressing Administered Items Approving updates to Administered Items Proposing Metadata Registry policies to the Executive Committee for approval Approving authorized Submitters, Read-only Users Approving Metadata Registry content, procedures, and formats. Acting on directions from the Executive Committee	Alter, Insert, Delete, Select
Steward	Ensuring that appropriate Administered Items Reviewing all Administered Items once they are in the "Recorded" status Ensuring the quality of metadata attribute values for Administered Items Proposing "Standard" registration status level Administered Items Proposing "Preferred Standard" registration status level Administered Items Recommending Submitters to the Registration Authority	Select, Insert
Submitter	Submitting Administered Items to the metadata registry Ensuring the completeness of mandatory metadata attributes	Insert, delete, select
Read-only user	Retrieval(review) the contents of the metadata register	Select

3.3 MCL Operator 정의

앞서 분석한 메타데이타 레지스트리에서 사용되는 연산 패턴 중 접근제어에 대해 표준 SQL 을 확장하여 MCL 로 정의하였다. MCL 은 사용자에 대하여 객체에 대한 접근 권한을 다루는 MGRANT 와 MREVOKE 라는 두 가지의 주요 연산자를 가진다. 권한을 부여하는 연산자를 MGRANT(GRANT FOR MDR)라고 정의하고, 권한을 회수하는 연산자를 MREVOKE(REVOKE FOR MDR)라고 정의하고 사용문법은 다음과 같다.

```
MGRANT USER_GROUP [ROLE] TO USER
MREVOKE USER_GROUP [ROLE] TO USER
```

3.4 MCL 사용 예제

이 절에서는 표 2, 표 3, 표 4 처럼 사용자와 사용자 그룹, 사용자 그룹과 역할, 역할과 권한의 관계를 표준 SQL 문을 이용한 방법과 이미 정의된 MCL 을 이용한 방법으로 비교한다.

표 2. 사용자와 사용자 그룹간의 관계

사용자	사용자 그룹
KIM	Read-only-users
SAM	Stewards

표 3. 사용자 그룹과 역할관계

사용자 그룹	Role
Read-only-users	데이터요소검색
Submitters	데이터요소검색
	데이터요소제출

표 4. 역할과 권한관계

Role	Privileges	Tables
데이터요소검색	Select	Data_elements
데이터요소제출	Select,Insert,Update,Delete	Data_elements

예제 1 은 데이타요소 검색 Role 을 Read_only_users 사용자 그룹에 부여하고 개별 사용자에게 권한을 부여하는 예이다. 예제 2 는 데이타요소 검색 Role, 데이터요소 제출 Role 을 Submitter 사용자 그룹에게 부여하고 개별 사용자에게 권한을 부여하는 예이다.

[예제 1] Kim 에게 Read_only_users 권한을 부여하시오

<SQL 의 DCL 방식>

(데이터요소검색)

- create role 데이터요소검색
- grant select on data_elements(table) to 데이터요소검색

(Read_only_users) Kim

- create role Read_only_users
- grant 데이터요소검색 to Read_only_users
- grant Read_only_users to Kim

< MCL 방식>

MGRANT read_only_users TO Kim

[예제 2] Sam 에게 Submitters 권한을 부여하시오

<SQL 의 DCL 방식>

(데이터요소검색)

- create role 데이터요소검색
- grant select data_elements(table) to 데이터요소검색

(데이터요소제출)

- create role 데이터요소제출
- grant insert, update data_elements(table) to 데이터요소제출

(Submitters) sam

- create role submitters
- grant 데이터요소검색, 데이터요소제출 to submitters
- grant submitters to sam

<SQL/MDR 의 MCL 방식>

MGRANT submitters TO sam

SQL 의 DCL 방식 사용시 개별 Role 들을 순차적으로 부여하는 복잡성과 이로 인한 잘못된 권한 부여로 접근시 문제가 있을 수 있다. 예제 1 과 예제 2 의 MCL 을 사용하면 개별 사용자에게 사용자 그룹 Role 을 바로 부여하는 방법을 사용하기 때문에 사용상의 편이성과 접근제어의 일관성을 유지할 수 있게 해준다.

4. 구현 및 평가

이 장에서는 MCL 을 위한 질의 처리기를 구현하고 결과에 대하여 기술한다. MCL 질의처리기는 구현의 용이성을 위해 기존 관계형 데이타베이스 관리 시스템을 바탕으로 하였다.

4.1 MCL Operator 구현 방법

MCL Operator 를 구현하는 방법으로는 GRANT 처럼 DBMS 에 MCL Operator 를 구현하는 방법과 ISO/IEC 11179 Part7 의 SQL/MM 처럼 User Define Type 으로 구현하는 방법이 있다. 2007 년 11 월 ISO/IEC JTC1/SC32 WG4 회의에서 신규기능 추가 시 User Define Type 으로 하기를 권장하고 있어 이 논문에서 구현은 User Define Type 으로 한다.

4.2 구현

4.2.1 구현 환경

ISO/IEC 11179 를 기준으로 하여 Role 기준정보와 사용자 그룹 기준정보를 그림 2 의 실선 구분을 미리 구성한다.

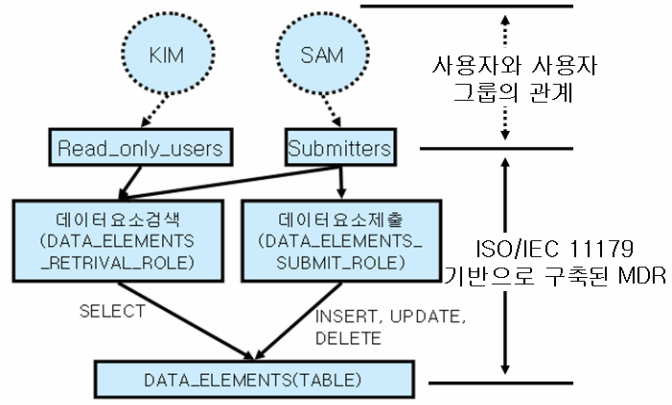

그림 2. ISO/IEC 11179 에 기준한 사용자 그룹정보

4.2.2 ISO/IEC 11179 Part 6 에 기반한 사전정보 구축

Role 과 권한과의 관계를 사전 정의된 내용을 테이블로 구성하면 표 4 와 같고, 사전 정의된 정보를 구성하는 과정은 아래와 같은 순서로 진행된다.

- create role 데이터요소검색
- grant select on data_elements (table) to 데이터요소검색
- create role 데이터요소제출
- grant insert, update, delete on data_elements (table) to 데이터요소제출

사용자 그룹과 Role 과의 관계도 사전 정의된 내용을 테이블로 구성하면 표 3 과 같이 되고 사전 정의된 정보를 구성하는 과정은 아래와 같은 순서로 진행된다.

- create role read_only_users
- grant 데이터요소검색 to Read_only_users
- create role submitters
- grant 데이터요소검색, 데이터요소제출 to Submitters

4.2.3 MCL 연산자를 이용한 사용자 권한 부여

ISO/IEC 11179 에서 정의된 사용자 그룹과 역할관계를 DBMS 로 미리 구성한 후 그림 3 과 같이 MCL 연산자로 실제 사용자에게 사용자 그룹에 대한 권한을 부여한다.

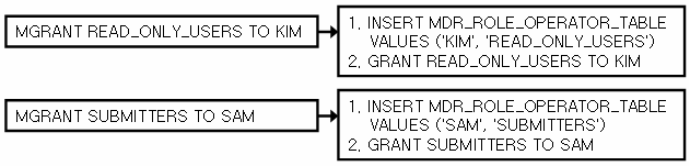

그림 3. MCL 처리시 DCL 처리내용

4.3 검증

검증단계에서는 부여한 권한에 대한 정확성에 대하여 확인 하도록 하겠다. 앞에서 부여한 권한을 보면 KIM 은 DATA_ELEMENTS 테이블에 대하여 SELECT 권한만을 가지고 SAM 은 DATA_ELEMENTS 테이블에 대하여 SELECT, INSERT, UPDATE, DELETE 권한을 가져야 한다. 그림 4, 그림 5 와 같이 DBMS 상에 부여된 권한을 확인하면 KIM 은 READ_ONLY_USERS 사용자 그룹에 속하고 DATA_ELEMENTS 테이블에 대하여 Select 권한을 가지고, SAM 은 SUBMITTERS 사용자 그룹에 속하고 DATA_ELEMENTS 테이블에 대하여 SELECT, DELETE, INSERT, UPDATE 권한을 가진다.

```
select USERNAME, y.USER_GROUP, GRANTED_ROLE, PRIVILEGE, TABLE_NAME
from
    (select TABLE_NAME, PRIVILEGE, a.GRANTED_ROLE, USER_GROUP
    from
        (select ROLE GRANTED_ROLE, TABLE_NAME, PRIVILEGE from ROLE_TAB_PRIVS) a,
        (select ROLE USER_GROUP, GRANTED_ROLE from ROLE_ROLE_PRIVS) b
    where a.GRANTED_ROLE=b.GRANTED_ROLE) x,
    (select USERNAME, GRANTED_ROLE USER_GROUP from USER_ROLE_PRIVS) y
where x.USER_GROUP=y.USER_GROUP;
```

USERNAME	USER_GROUP	GRANTED_ROLE	PRIVILEGE	TABLE_NAME
KIM	READ_ONLY_USERS	DATA_ELEMENTS_RETRIVAL_ROLE	SELECT	DATA_ELEMENTS

그림 4. RDB 상에 부여된 KIM 의 권한

```
select USERNAME, y.USER_GROUP, GRANTED_ROLE, PRIVILEGE, TABLE_NAME
from
    (select TABLE_NAME, PRIVILEGE, a.GRANTED_ROLE, USER_GROUP
    from
        (select ROLE GRANTED_ROLE, TABLE_NAME, PRIVILEGE from ROLE_TAB_PRIVS) a,
        (select ROLE USER_GROUP, GRANTED_ROLE from ROLE_ROLE_PRIVS) b
    where a.GRANTED_ROLE=b.GRANTED_ROLE) x,
    (select USERNAME, GRANTED_ROLE USER_GROUP from USER_ROLE_PRIVS) y
where x.USER_GROUP=y.USER_GROUP;
```

USERNAME	USER_GROUP	GRANTED_ROLE	PRIVILEGE	TABLE_NAME
SAM	SUBMITTERS	DATA_ELEMENTS_RETRIVAL_ROLE	SELECT	DATA_ELEMENTS
SAM	SUBMITTERS	DATA_ELEMENTS_SUBMIT_ROLE	DELETE	DATA_ELEMENTS
SAM	SUBMITTERS	DATA_ELEMENTS_SUBMIT_ROLE	INSERT	DATA_ELEMENTS
SAM	SUBMITTERS	DATA_ELEMENTS_SUBMIT_ROLE	UPDATE	DATA_ELEMENTS

그림 5. RDB 상에 부여된 SAM 의 권한

사용자 KIM 과 SAM 의 권한에 대하여 검증하기 위해 질의문 Q1, Q2, Q3, Q4 를 다음과 같이 만들었다.

Q1) SELECT REG_AUTH_IDENTIFIER, DATA_IDENTIFIER, VERSION, CREATION_USER FROM MDR.DATA_ELEMENTS;

Q2) UPDATE MDR.DATA_ELEMENTS SET CREATION_USER='KIM'
WHERE REG_AUTH_IDENTIFIER=200800001
AND DATA_IDENTIFIER=200804001 AND VERSION=1 ;

Q3) INSERT INTO MDR.DATA_ELEMENTS(REG_AUTH_IDENTIFIER, DATA_IDENTIFIER,VERSION,
RESPONSIBILITY_NAME, CREATION_USER)
VALUES(200800002, 200804002, 2, 'DEFAULT', 'SAM');

Q4) DELETE FROM MDR.DATA_ELEMENTS
WHERE REG_AUTH_IDENTIFIER=200800002
AND DATA_IDENTIFIER=200804002 AND VERSION=2;

```
SQL> CONN KIM/KIM
연결되었습니다.
SQL> SELECT REG_AUTH_IDENTIFIER, DATA_IDENTIFIER, VERSION, CREATION_USER
  2  FROM MDR.DATA_ELEMENTS;

REG_AUTH_IDENTIFIER DATA_IDENTIFIER    VERSION CREATION_USER
------------------- ---------- ---------- ------------------------------
          200800001  200804001          2 KIM

SQL> UPDATE MDR.DATA_ELEMENTS SET CREATION_USER='KIM'
  2  WHERE REG_AUTH_IDENTIFIER=200800001
  3        AND DATA_IDENTIFIER=200804001 AND VERSION=1;
UPDATE MDR.DATA_ELEMENTS SET CREATION_USER='KIM'
       *
1행에 오류:
ORA-01031: 권한이 불충분합니다


SQL> INSERT INTO MDR.DATA_ELEMENTS(REG_AUTH_IDENTIFIER, DATA_IDENTIFIER,
  2                                VERSION, RESPONSIBILITY_NAME, CREATION_USER)
  3  VALUES(200800002, 200804002, 2, 'DEFAULT', 'SAM');
INSERT INTO MDR.DATA_ELEMENTS(REG_AUTH_IDENTIFIER, DATA_IDENTIFIER,
            *
1행에 오류:
ORA-01031: 권한이 불충분합니다


SQL> DELETE FROM MDR.DATA_ELEMENTS
  2  WHERE REG_AUTH_IDENTIFIER=20080002
  3        AND DATA_IDENTIFIER=200804002 AND VERSION=2;
DELETE FROM MDR.DATA_ELEMENTS
            *
1행에 오류:
ORA-01031: 권한이 불충분합니다
```

그림 6. Q1, Q2, Q3, Q4 에 대한 KIM 의 권한

DATA_ELEMENTS 테이블에 대하여 SELECT 권한만 가지는 사용자 KIM 에 대한 Q1, Q2, Q3, Q4 실행결과 그림 6 과 같이 Q1 은 정상 실행이 되고, Q2, Q3, Q4 는 권한 오류 에러가 정상적으로 발생하였다.

```
SQL> CONN SAM/SAM
연결되었습니다.
SQL> SELECT REG_AUTH_IDENTIFIER, DATA_IDENTIFIER, VERSION, CREATION_USER
  2  FROM MDR.DATA_ELEMENTS;

REG_AUTH_IDENTIFIER DATA_IDENTIFIER    VERSION CREATION_USER
------------------- ---------- ---------- ------------------------------
          200800001  200804001          1 KIM

SQL> UPDATE MDR.DATA_ELEMENTS SET CREATION_USER='KIM'
  2  WHERE REG_AUTH_IDENTIFIER=200800001
  3        AND DATA_IDENTIFIER=200804001 AND VERSION=1;

1 행이 갱신되었습니다.

SQL> INSERT INTO MDR.DATA_ELEMENTS(REG_AUTH_IDENTIFIER, DATA_IDENTIFIER,
  2                                VERSION, RESPONSIBILITY_NAME, CREATION_USER)
  3  VALUES(200800002, 200804002, 2, 'DEFAULT', 'SAM');

1 개의 행이 만들어졌습니다.

SQL> DELETE FROM MDR.DATA_ELEMENTS
  2  WHERE REG_AUTH_IDENTIFIER=200800002
  3        AND DATA_IDENTIFIER=200804002 AND VERSION=2;

1 행이 삭제되었습니다.
```

그림 7. Q1, Q2, Q3, Q4 에 대한 SAM 의 권한

DATA_ELEMENTS 테이블에 SELECT, UPDATE, INSERT, DELETE 을 가지는 사용자 SAM 에 대하여 Q1, Q2, Q3, Q4 실행결과 그림 7 과 같이 Q1, Q2, Q3, Q4 가 권한 오류 없이 정상적으로 실행되었다.

4.4 비교평가

DCL 과 MCL 의 특징을 비교해 보면 표 5 와 같이 MCL 은 처리구조가 간단하여 사용이 편하고 미리 정의된 구조를 사용하므로 보안성이 우수하다.

표 5. DCL 과 MCL 비교

항 목	DCL	MCL
사용자 편이성	어렵다	쉽다
보안성	복잡한 처리구조로 인하여 접근보안성에 대한 검증 필요	우수함
접근방법의 표준화	미지원	지원
질의문의 간편성	복 잡	간편
MDR 이해도	정확하고 높은 이해도 필요	낮음
MDR 접근제어 상호 운용성	낮음	높음

5. 결론

MDR 은 메타데이타의 관리 및 상호 운용성 향상을 위하여 개발된 ISO/IEC 11179 의 핵심요소이며, 다양한 분야에서 MDR 시스템들이 개발되었다. 그러나 시스템들이 MDR 표준을 따르지 않고 개발되거나 구축되어 MDR 간 메타데이타 불일치가 발생하고 MDR 간에 데이터를 공유하고 교환할 수 있는 표준화된 연산자가 없어 개발 비용 및 시간이 많이 드는 문제가 있었다.

이러한 문제를 해결하기 위해 SQL/MDR 이 제안되었으나 SQL/MDR 은 검색을 위한 연산만을 지원하고 있다. 즉, 메타데이타에 대한 수정, 삭제, 추가는 물론 구조의 일관성 있는 정의 및 접근제어를 위한 연산은 제공하지 않는다. 특히 접근제어는 안전하고 정형화 된 MDR 에 대한 보안을 위해 필수적으로 지원되어야 한다.

이 논문에서는 ISO/IEC 11179 Part 6 에서 제안하는 사용자 그룹별 권한과 역할을 미리 구축하고 MCL 이라는 연산자를 제공하여 시스템 관리자가 표준 및 저장구조에 대한 명확한 이해와 사용자 권한에 대한 이해 없이도 접근 시스템을 쉽고 안전하게 개발할 수 있도록 했다. 먼저 MCL 의 개념 및 정의에 대해서 기술하고 MDR 사용자 그룹의 Role 을 검토하여 MCL Operator 를 정의하였다. 또한 DCL 과 MCL 방법을 비교하고 MCL Operator 를 구현하고 검증을 하였다.

향후 연구로서, 실제 MDR 구축을 위한 MDR 조작을 위한 언어 및 MDR 구조를 생성하는 언어에 대한 연구가 필요하다.

참고문헌

[1] ISO/IEC JTC 1/SC 32, “ISO/IEC 11179: Specification and standardization of data elements, Part 1 ~ 6,” 2003.

[2] 신동길, 김영갑, 정동원, 박수현, 백두권, “메타데이타 레지스트리의 일관성 있는 접근을 위한 질의 언어”, 정보과학회 논문지, 데이타베이스 제 31권 제6호, p609-623, 2004.12.

[3] Dong-won Jeong, Young-Gab Kim, and Hoh Peter In, “Quantitative Evaluations on the Query Modeling and System Integrating Cost of SQL/MDR,” ETRI Journal, Volume, Number 4, p367-376, August 2005.

[4] ISO/IEC JTC 1/SC 32/WG 3, “ISO/IEC 9075, Database Language SQL3, Part1 ~ 10,” 1999.

[5] ETRI, Research on the Registration and Search System of Component, Research Report, 2002.

[6] KISTI, A study on the development of standardization and management model for science and technology information, Research Report, 2002.

[7] EPA, Environmental Data Registry, http://www.epa.gov/edr/

[8] EPA, “Data Standards Publications and Guidances,” 2003.

[9] AIHW, Australian National Health Information Knowledgebase, http://www.aihw.gov.au/

[10] Australian National Health Data Committee, “National Health Data Dictionary,” 2003.

[11] U.S. Transportation System, http://www.dot.gov/.

[12] ITS Architecture Development Team, “ITS Logical Architecture-Volume I, Volume II: Process Specifications, Volume III: Data Dictionary,” 2002.

[13] Egenhofer, M. “Spatial SQL: A query and presentation language,” IEEE Transactions on Knowledge and Data Engineering, Vol. 6, No. 1, pp.86-95, 1994.

[14] Lee, J.-Y., “Integrating Spatial and Temporal Relationship Operators into SQL3 for Historical Data Management,” ETRI Journal, Vol.24, No.3, pp. 226-238, 2002.

[15] Pissinou, N., Snodgrass, R., Elmasri, R., Mumick, I., Ozsu, T., Pernici, B., Segev, A., Theodoulidis, B., and Dayal, U., “Towards an Infrastructure for Temporal Databases: Report of an Invitational ARPA/NSF Workshop,” Vol. 23, No. 1, pp. 35-51, In SIGMOD Record, 1994.

[16] ISO/IEC JTC 1/SC 32, “ISO/IEC 13249: Information technology- Database languages- SQL Multimedia and Application Packages,” 2003.