

# 메타데이터 레지스트리의 일관성 있는 접근을 위한 질의 언어

(A Query Language for Consistent Access of Metadata Registries)

신 동 길 <sup>†</sup>    김 영 갑 <sup>†</sup>    정 동 원 <sup>\*\*</sup>    박 수 현 <sup>\*\*\*</sup>    백 두 권 <sup>\*\*\*\*</sup>  
 (Dongkil Shin) (Young-Gab Kim) (Dongwon Jeong) (Soo-Hyun Park) (Doo-Kwon Baik)

**요 약** 현재 세계 여러 국가에서 다양한 메타데이터 레지스트리들이 구축되어 왔다. 그러나 메타데이터 레지스트리가 국제 표준 (ISO/IEC 11179)임에도 불구하고 일관성 있는 표준 접근 인터페이스를 제공하지 않아 MDR 시스템 마다 다른 연산으로 구축되었다. 이는 메타데이터 레지스트리 구축시 동일한 연산 패턴에 대한 중복된 노력으로 인한 불필요한 시간과 비용의 낭비를 야기한다. 또한 개발된 MDR 시스템 마다 MDR 구성요소들에 대한 상이한 인터페이스를 이용함으로써 MDR간 불일치를 발생시키고 있다. 이 논문에서는 MDR 시스템에서 공통적으로 사용되는 연산 패턴을 분석하고 정의하며 분석된 연산 패턴을 이용하여 표준 SQL을 확장한 SQL/MDR을 정의하고 설계한다. SQL/MDR은 MDR 관리 시스템 개발에 표준 접근 인터페이스로 제공하며 SQL/MDR을 이용하여 서지 데이터베이스에 실제 적용하고 구현한 결과를 기술한다. SQL/MDR을 이용하여 MDR 시스템을 개발할 경우 불필요한 노력과 시간을 줄이고 메타데이터 레지스트리에 대한 일관성 있는 접근을 가능하게 하며 ISO/IEC 11179의 명세를 준수하도록 유도할 수 있다.

**키워드** : MDR, 메타데이터 레지스트리, SQL, ISO/IEC 11179, 데이터요소, 메타데이터 레지스트리 질의 언어

**Abstract** Various metadata registries have been built in many countries of the world. Although the metadata registry is an international standard, it does not provide a consistent access interface for handling the metadata registries. Therefore, all systems for managing them were developed by using different operations and access interfaces. It requires duplicate efforts on the same operations whenever metadata registry systems are developed. As a result, it causes unnecessary costs and efforts for building metadata registries, and also incurs inconsistency between the metadata registries because the previous developed systems use the different interfaces for the metadata registry elements. This paper analyzes and defines operation patterns that are commonly used for the metadata registries. We defined and designed SQL/MDR extended from SQL using the analyzed operation patterns. SQL/MDR provides a standardized access interface for developing metadata registry systems. This paper shows the implementation of SQL/MDR and the result that we actually applied it to the bibliographical databases. By developing the metadata registry systems using SQL/MDR, we can reduce much time and efforts owing to its standard interface. It allows metadata registries to be accessed consistently. Additionally, it makes all metadata registries follow the international standard, ISO/IEC 11179.

**Key words** : MDR, Metadata Registry, SQL, ISO/IEC 11179, Data Element, Metadata Registry Query Language

<sup>†</sup> 비 회 원 : 고려대학교 컴퓨터학과  
 dkshin@software.korea.ac.kr  
 ygkim@software.korea.ac.kr

<sup>\*\*</sup> 정 회 원 : 고려대학교 컴퓨터학과 교수  
 withimp@korea.ac.kr

<sup>\*\*\*</sup> 정 회 원 : 국민대학교 비즈니스IT학부 교수

shpark21@kookmin.ac.kr

<sup>\*\*\*\*</sup> 종신회원 : 고려대학교 컴퓨터학과  
 baik@software.korea.ac.kr

논문접수 : 2004년 2월 13일

심사완료 : 2004년 9월 14일

## 1. 서론

메타데이터 레지스트리를 기반으로 한 통합방법에 대한 연구가 활발해짐에 따라 많은 메타데이터 레지스트리(Metadata Registry: MDR)와 이를 관리하기 위한 시스템들이 개발되어 왔다. 메타데이터 레지스트리는 메타데이터의 동적 관리 및 상호운용성 향상을 위해 ISO/IEC JTC 1에 의해 개발된 국제 표준으로 데이터 표준화, 데이터의 메타정보 검색과 등록, 데이터에 대한 이해와 공유향상 등의 기능을 제공한다. 이러한 장점 때문에 현재까지 국내·외 많은 기관들이 메타데이터 관리를 위해 ISO/IEC 11179를 기반으로 하여 메타데이터 레지스트리를 구축해 왔다.

MDR 시스템을 개발하기 위해서는 ISO/IEC 11179를 준수해야 한다. ISO/IEC 11179에서는 MDR의 개발 및 관리를 위해 데이터요소 검색, 데이터요소 등록, 데이터요소 투표, 데이터요소 등록상태 변경, 데이터요소 개정 및 버전관리 등과 같은 다양한 기능에 대하여 명세하고 있다.

ISO/IEC 11179 표준에 따라 개발된 MDR 시스템은 크게 두 가지 특징을 갖는다. 첫째, MDR 접근시 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 11179를 준수하여 시스템 개발시 명세에서 요구하는 필수 테이블과 필수 속성이 있다.

현재까지 구축된 MDR들은 몇 가지 문제를 지닌다. 첫째, MDR 시스템 접근시 사용되는 동일한 연산 패턴들에 대한 일관성 있는 접근방법이 연구되지 않았다. 이는 MDR 시스템 개발시 ISO/IEC 11179를 준수하기 위해 매년 동일한 연산 패턴을 다른 방법으로 개발함으로써 시간과 비용의 낭비를 초래한다. 또한 표준 접근 방법을 제공하지 않기 때문에 MDR 접근시 반드시 요구되는 제약조건이 고려되지 않는 경우가 발생한다. 이는 MDR간의 이질성 문제를 야기하고 표준화된 MDR 구축을 어렵게 한다. 둘째, ISO/IEC 11179에서는 기능에 대한 개략적인 설명만 있을 뿐 구체적인 스키마를 제공하지 않는다. 따라서 개발된 MDR 시스템마다 다른 구조와 명칭을 갖는 스키마를 사용하며 이는 결국 MDR간의 이질성을 유발한다.

이 논문에서는 이를 위해 먼저 ISO/IEC 11179에서 요구하고 있는 테이블과 속성을 정의한다. 또한 동일한 패턴을 지니는 접근연산들을 분석하고 일관성 있는 접근을 위한 MDR 질의 언어를 정의 및 설계한다. 국제 표준 질의어인 SQL[1]을 기반으로 하여 확장된 메타데

타 레지스트리 질의 언어를 SQL/MDR이라고 정의하며 ISO/IEC 11179의 기능적 특성을 포함한 연산자 이름을 정의함으로써 MDR 시스템 개발을 보다 용이하게 한다.

SQL/MDR은 ISO/IEC 11179를 준수하는 MDR 구축을 가능하게 하고 일관성 있는 접근방법을 제공하기 때문에 MDR 관리 시스템 개발, 질의 모델링, 분산 MDR 통합 등에 있어 비용과 시간을 감소시키게 되며, 보다 단순화된 상호 메타데이터 교환 메커니즘을 가능하게 한다.

## 2. 관련연구

ISO/IEC 11179는 데이터를 쉽게 이해하고 공유할 수 있도록 하기 위해 국제 표준 기구인 ISO/IEC JTC 1에 의해 개발되었다. 이는 데이터베이스 간 공유 및 교환을 극대화하고 일관성 있게 데이터베이스를 표현하고 구축하기 위한 국제 표준이다. ISO/IEC 11179는 데이터 요소라는 중요한 개념을 지니고 있으며, 이는 정의, 식별, 표현 및 허용 가능한 값들의 속성 집합으로 데이터를 명세하기 위한 최소단위이다. 이러한 데이터 요소들의 집합을 메타데이터 레지스트리라 한다[2].

현재 국내에서 구축된 메타데이터 레지스트리는 한국 전자통신연구원의 컴포넌트의 원활한 유통을 위해 구축한 컴포넌트 메타데이터 레지스트리[3], 한국과학기술정보 연구원의 서지정보 메타데이터 레지스트리[4]가 있다. 해외에서 구축된 메타데이터 레지스트리는 미국 환경보호협회의 EDR(Environmental Data Registry)[5, 6], 호주의 NHIK(Australian National Health Information Knowledgebase)[7,8], 미국의 ITS(Intelligent Transportation System)[9,10] 등이 있다.

EDR은 환경 데이터에 대한 참조 정보로서 정의, 자원, 그리고 환경 데이터의 사용에 대한 포괄적이고 권위 있는 참조 정보를 가지고 있다. NHIK는 건강과 관련된 메타데이터에 대한 전자 저장소(Electronic Repository)이며 질의 도구이다. ITS는 미 교통부에서 50개 주와 지역의 교통국이 ITS 시스템 내부에서 혹은 중앙정부와 ITS 외부 시스템이 가지고 있는 데이터를 의미에 따라 교환하고 상호작용 할 수 있도록 하는 메타데이터 레지스트리로 구축된 시스템이다.

이들은 모두 국제 표준인 ISO/IEC 11179를 기반으로 구축되었음에도 불구하고 동일한 접근연산을 위해 서로 상이한 접근 방법으로 개발하여 이용하고 있다. 즉, 일관성과 표준화가 결여된 메타데이터 레지스트리 접근 방법을 사용한다. 이는 불필요한 노력과 비용 낭비를 초래하게 되며, 서로 다른 접근 방법을 사용함으로써 국제

표준인 MDR 국제표준 규격에 위배된다. 또한 MDR 접근시 반드시 요구되는 제약조건을 고려하지 않아 MDR 간 이질성을 야기함으로써 국제표준인 ISO/IEC 11179의 혼란을 발생시킨다.

따라서 메타데이터 레지스트리에 대한 표준 인터페이스를 제공하기 위한 일관성 있는 접근 방법에 대한 연구가 필요하다. 이미 MDR과 같이 특수한 많은 데이터베이스 분야에서는 이러한 연구가 진행되어 왔다. 즉, 공간 데이터, 시간 데이터 및 멀티미디어 데이터 등에 대한 일관성 있는 접근을 위한 표준 인터페이스 개발되었다. 공간 데이터의 표준 접근을 위한 공간 질의어(Spatial Query Language) 연구[11,12], 시간 데이터를 위한 시간 질의어(Temporal Query Language) 연구[12,13], 멀티미디어 데이터를 위한 SQL/MM[14] 등이 그 예이며, 이들은 모두 국제 표준 질의 언어인 SQL[1]을 기반으로 확장된 질의 언어들이다.

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

### 3. 메타데이터 레지스트리 접근 패턴

메타데이터 레지스트리 연산자를 정의하기 위해서는 메타데이터 레지스트리에 대한 접근연산 패턴을 먼저 정의해야 한다. 이를 위해 이 논문에서는 ISO/IEC 11179의 기능과 필수 구성요소 및 기존 MDR 시스템들에서의 실질적인 접근 패턴을 분석한다. 이 장에서는 분석 결과를 통해 정의된 접근연산 패턴에 대하여 기술한다.

#### 3.1 연산자 정의를 위한 분석 방법

이 절에서는 접근연산 패턴을 정의하기 위한 분석 방법에 대하여 기술하고 분석 결과를 요약한다. 이 논문에서는 데이터요소들을 논리적으로 그룹핑하는 구성요소들을 그룹핑요소라고 정의하며 데이터요소와 그룹핑요소를 각각 정의하면 다음과 같다.

**정의 1.** 데이터요소 : 데이터의 특성을 나타내는 일반적이고 추상적인 개념으로 하나의 데이터를 나타내기 위한 메타데이터이다.

**정의 2.** 그룹핑요소 : 데이터요소를 특정 기준으로 분류한 논리적 단위. 그룹핑요소에는 데이터요소개념, 개념영역, 객체 클래스가 있다.

메타데이터 레지스트리에 대한 접근연산 패턴들에 대하여 정의하기 위해 ISO/IEC 11179에서의 주요 기능을 위해 요구되는 연산 패턴과 필수 구성요소들에 대한 접근을 위해 요구되는 연산 패턴들을 분석한다. 또한 실질적으로 이용되고 있는 접근연산 패턴들을 추출하기 위해 기존 MDR 시스템들에서 제공하는 연산들에 대하여 분석한다. 그림 1은 이와 같은 분석 방법을 도식적으로 보여준다.

그림 1에서 보여주듯이, 분석 과정은 ISO/IEC 11179의 메타모델 분석과 기존 MDR시스템의 분석 과정으로 분류되며 두 분석 과정을 통해 얻어진 연산 패턴들을 기반으로 메타데이터 레지스트리 연산자를 정의하게 된다.

ISO/IEC 11179에서 가장 중요한 구성요소는 데이터요소이며, 대부분의 연산들이 데이터요소를 중심으로 이루어진다. 즉, 이는 메타데이터 레지스트리에 대한 접근시 대상 객체가 대부분 데이터요소가 된다는 것을 의미한다. 데이터요소와 더불어 ISO/IEC 11179에서의 필수 구성요소로는 개념 도메인(Conceptual Domain), 데이터요소 개념(Data Element Concept) 및 객체 클래스(Object Class) 등이 있으며 이들은 데이터요소를 논리적으로 그룹핑하는 그룹핑요소이다. 따라서 데이터요소

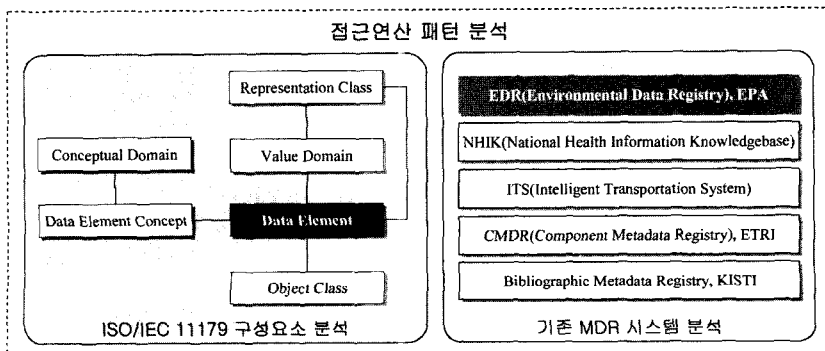


그림 1 접근연산 패턴 분석

에 대한 접근연산과 함께 그룹핑요소에 대한 접근연산이 가능하다. 그러므로 이들에 대한 메타데이터 레지스트리 연산자들의 정의가 요구된다. 추가적으로 그룹핑요소는 데이터요소를 논리적으로 그룹핑하는 단위이므로 그룹핑요소와의 연관성을 이용한 데이터요소 접근연산이 가능하다. 이는 이러한 접근연산 패턴들에 대한 연산자를 제공해야함을 의미한다.

기존 MDR 시스템 분석을 위해 국내·외 대표적인 MDR 시스템들을 선정하였다. 대부분의 MDR 시스템들은 데이터요소에 대한 접근연산만을 제공하며, 일부 시스템에서 그룹핑요소에 대한 검색과 같은 접근연산을 제공하고 있다. 기존 MDR 시스템들에 대한 분석 결과, 실질적으로 제공되고 있는 접근연산 패턴들이 첫 번째 분석 과정을 통해 추출된 연산 패턴의 일부만을 사용하고 있음을 알 수 있다.

결론적으로, 이 논문에서는 보다 다양한 검색 기능을 위해서 기존 MDR 시스템의 모든 검색연산을 수용하고 ISO/IEC 11179에서 요구하는 모든 필수 속성별 검색연산을 포함시킴으로써 보다 다양한 검색이 가능하도록 하였다. 즉, 선정된 연산자는 기존 MDR 시스템에서 실제 사용되고 있는 모든 검색연산 패턴을 수용하고 있으며, 추가적인 보다 다양한 연산자들을 제공한다. 따라서 ISO/IEC 11179에 대한 분석을 통해 정의된 연산 패턴을 기반으로 메타데이터 레지스트리 연산자를 정의할 경우, 실질적인 응용 및 활용에 있어 충분한 연산자들을 제공할 수 있음을 알 수 있다. 또한 향후 추가적인 접근연산 패턴들이 요구될 경우, 이러한 연산 패턴을 위한 메타데이터 레지스트리 연산자를 정의하고 추가함으로

써 추출되지 않은 연산들을 수용할 수 있다.

**3.2 메타데이터 레지스트리 기본연산**

ISO/IEC 11179는 메타데이터 레지스트리에 대한 프레임워크를 제공한다. ISO/IEC 11179에서 명세하고 있는 프레임워크는 기본적으로 준수해야만 하는 다양한 기능을 요구한다. ISO/IEC 11179에서의 MDR 구축과정은 그림 2와 같다.

ISO/IEC 11179에 명시된 메타데이터 레지스트리 관리를 위한 기본적인 기능은 데이터요소 검색, 데이터요소 등록, 데이터요소 투표, 데이터요소 등록상태 관리 및 데이터요소 개정 등이 있다. 이 프로세스를 통해 MDR 시스템은 지속적으로 확장, 유지 및 관리된다. 표 1은 MDR 시스템의 기능을 간략하게 요약한 내용이다.

데이터요소 검색은 MDR 시스템에서 가장 중요한 기능으로 사용자가 이 기능을 이용하여 MDR 시스템에 등록되어 있는 모든 데이터요소를 검색할 수 있다. 데이터요소 검색은 많은 선택옵션을 사용하며 모든 MDR 시스템 기능 중 사용빈도 및 중요도가 가장 높은 기능이다. 또한 데이터요소 검색 기능에서 사용되는 MDR 시스템 접근연산은 다른 기능에 비해 정형화된 연산을 사용하여 패턴 추출이 매우 용이하다. 사용자는 데이터요소 검색 기능을 이용하여 자신이 찾는 분야에서 원하는 데이터요소를 찾아 해당 요소의 메타정보를 확인할 수 있다. 또한 연산은 정의, 이름, 클래스, 개념영역, 개념, 등록상태 검색 옵션을 갖는다.

데이터요소 등록은 데이터요소 검색을 통해 사용자가 검색하고자 하는 데이터요소가 없을 경우, 해당 MDR 시스템의 Steward와 Registrar를 통해 새로운 데이터요

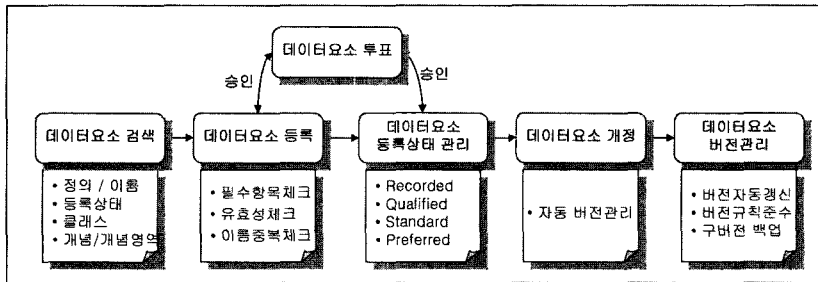


그림 2 ISO/IEC 11179에서의 MDR 구축과정

표 1 MDR 시스템 기능 요약

구분	데이터요소 검색	데이터요소 등록	등록상태 관리	버전관리
기능 및 제약 조건	<ul style="list-style-type: none"> <li>• 데이터요소 정의/이름별 검색</li> <li>• 데이터요소 등록상태별 검색</li> <li>• 데이터요소 클래스별 검색</li> <li>• 데이터요소 개념별 검색</li> <li>• 데이터요소 개념영역별 검색</li> <li>• 클래스/개념/개념영역 검색</li> </ul>	<ul style="list-style-type: none"> <li>• 필수항목검사</li> <li>• 속성들의 정보 유효성검사</li> <li>• 데이터요소 이름중복검사</li> </ul>	<ul style="list-style-type: none"> <li>• Candidate</li> <li>• Recorded</li> <li>• Qualified</li> <li>• Standard</li> <li>• Preferred</li> <li>• Retired</li> </ul>	<ul style="list-style-type: none"> <li>• 데이터요소 개정에 따른 버전관리                             <ul style="list-style-type: none"> <li>- 버전 자동갱신</li> <li>- 버전 규칙준수</li> <li>- 구버전 자동백업</li> </ul> </li> </ul>

소를 등록하는 기능이다. MDR은 데이터요소 등록 기능을 통해 지속적으로 확장되며 등록된 데이터요소는 데이터요소 투표 기능의 Control Committee 승인과정을 거쳐 정식 데이터요소로 채택된다. 데이터요소 등록시 데이터요소 이름은 유일해야 하므로 중복검사 연산 등을 한다.

데이터요소 등록 상태관리는 등록된 데이터요소의 상태를 투표를 통해 순차적으로 관리하는 기능이다. 데이터요소는 'Candidate', 'Recorded', 'Qualified', 'Standard', 'Preferred' 및 'Retired'의 6개의 등록상태를 갖는다. 데이터요소 등록 기능을 통해 등록된 요소는 투표 절차를 거쳐 'Qualified' 등록상태를 갖으며 사용빈도 및 중요도에 따라 데이터요소 투표 기능을 통해 표준 상태로 단계적인 승격이 이루어진다.

데이터요소 개정은 Control Committee가 데이터요소의 개정여부를 판단해 데이터요소의 메타정보 일부를 수정하는 기능이다. 데이터요소 개정을 통해 MDR 시스템에 등록되어 있는 데이터요소는 정제화 된다. 데이터요소가 개정되면 개정된 데이터요소의 버전은 갱신되며 데이터요소 버전관리 기능을 통해 버전이 관리 된다.

3.3 ISO/IEC 11179의 필수 구성요소

ISO/IEC 11179에서는 메타데이터 레지스트리에 대한 메타모델을 제공한다. 메타모델은 그림 3과 같이 크게 6개의 필수 구성요소로 구성되어 있다. 메타모델의 각 구성요소들은 데이터요소를 중심으로 MDR 시스템을 위한 메타모델을 나타내고 있다. ISO/IEC 11179에서 정의하고 있는 메타모델은 모든 MDR 시스템이 준수하는 표준으로 이 논문에서는 그림 3의 메타모델을 기반으로 연산 패턴을 선정하였다.

메타모델의 구성요소는 그룹핑요소와 데이터요소로 분류되며 그룹핑요소에는 개념 영역(Conceptual Domain), 데이터 요소 개념(Data Element Concept), 개체 클래스(Object Class)가 있다. 메타데이터 레지스트리에 대한 접근연산 결과는 두 요소의 집합이 되며 가장 일반적인 접근 대상은 데이터요소이다. 데이터요소는

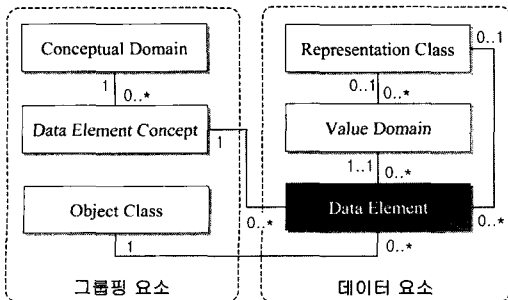


그림 3 ISO/IEC 11179 메타모델

ISO/IEC 11179에서 필수속성으로 정한 Name, Definition, Context, Registration Status(Standard, Recorded 등) 등을 이용한 검색 연산이 가능하다. 또한 데이터요소는 여러 등록 상태를 지니므로 이를 이용한 검색 연산 패턴 또한 수용해야 한다. 그룹핑요소는 데이터요소에 대한 논리적인 그룹핑 단위이다. 따라서 이러한 데이터요소와 그룹핑요소와의 관계성을 이용한 데이터요소 검색 연산 패턴이 지원되어야 한다. 마지막으로, 그룹핑요소를 검색하기 위한 연산 패턴 또한 연산자로서 제공되어야 한다.

이와 같은 분석 결과를 바탕으로 접근연산 패턴을 다음과 같이 분류할 수 있다.

- 데이터요소 검색(접근 대상 = 데이터요소)
  - 등록상태를 이용한 검색 (데이터요소의 생명주기를 이용한 검색)
  - 필수 속성을 이용한 검색
  - 그룹핑요소 한정을 통한 데이터요소 검색 (그룹핑요소와 관계성을 이용한 검색)
- 그룹핑요소 검색(접근 대상 = 그룹핑요소)
  - 필수 속성을 이용한 검색

3.4 기존 MDR 시스템 분석

이 절에서는 EDR, KISTI, ETRI, ITS 및 NHIK 등 기존 MDR 시스템에서 실질적으로 제공하는 연산 패턴들에 대한 분석 결과에 대하여 기술한다. 특히, 이 논문에서는 기존 시스템들 중에서 가장 다양하고 체계적인 서비스를 제공하는 미 환경청의 EDR을 중심으로 분석한 연산 패턴들에 대하여 서술한다.

EDR은 ISO/IEC 11179를 가장 잘 준수하고 있는 대표적인 시스템이다. EDR의 검색패턴 분석을 통해 연산자의 실제 MDR 시스템에서 운영되고 있는 접근연산을 살펴보면 EDR의 데이터요소 검색 패턴은 그림 4와 같다.

EDR은 방대한 양의 데이터요소 검색을 위해 그림과 같이 여러 종류의 검색패턴을 갖고 있다. 검색메뉴는 이름, 정의, 객체 클래스 검색 등이 있으며 각각의 검색 옵션은 포함(Containing), 시작(Beginning With), 일치(Exact Match) 검색이 있다.

이 논문에서는 EDR의 검색패턴과 기타 MDR 시스템

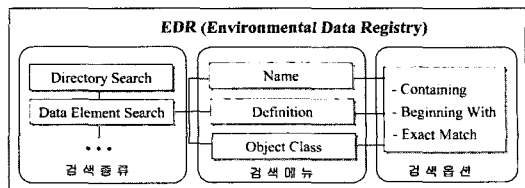


그림 4 EDR의 데이터요소 검색 패턴

에서 사용되고 있는 검색패턴을 모두 수용하며 ISO/IEC 11179에서 요구하는 모든 필수 속성으로 검색할 수 있도록 가능한 모든 연산자를 포함하였다.

**3.5 연산 패턴 분석 결과**

검색에 사용되는 연산 패턴은 검색 대상에 따라 크게 데이터요소 검색과 그룹핑요소 검색으로 분류된다. 이런 연산 패턴은 모든 MDR 시스템에서 동일한 데이터요소 검색패턴을 갖는다. 검색연산은 표 2와 같이 데이터요소 검색연산과 그룹핑요소 검색연산으로 이루어져 있다. 사용자는 검색연산들과 해당 옵션을 조합해 원하는 데이터요소 및 그룹핑요소 항목을 검색한다. 특히 검색 연산은 다양한 형태의 연산을 하기 때문에 각각의 연산을 명시적인 이름을 갖는 연산자로 새로 정의함으로써 ISO/IEC 11179를 정확히 이해하지 못한 개발자에게 MDR 시스템 개발시 유용성을 제공한다.

데이터요소 검색을 통해 사용자가 원하는 데이터요소가 없을 경우 사용자는 데이터요소를 등록한다. 이때, 등록되는 데이터요소 이름은 반드시 이미 등록된 다른 데이터요소 이름과 중복되지 않는 이름을 입력해야 한다. 이는 메타데이터 레지스트리에 대한 데이터요소 등록연산 수행시, 반드시 중복여부를 검사해야 함을 의미한다. MDR 시스템 개발시 미리 정의된 SQL/MDR을 이용하면 시간을 절약할 수 있으며 개발시 중복여부를 자동으로 체크할 수 있다.

데이터요소 상태 관리를 통해 데이터요소는 Control Committee의 데이터요소 투표결과에 따라 등록상태가 'Candidate'에서 'Recorded', 'Qualified', 'Standard', 'Preferred' 및 'Retired'로 갱신된다. 등록상태가 갱신될 때마다 등록상태와 관련된 메타정보가 함께 자동으로

수정되는 'UPDATE' 연산이 발생한다. 이 연산 패턴을 이용하여 데이터요소의 상태 단계별로 수정되는 항목을 자동으로 처리할 수 있다.

데이터요소 등록 기능을 통해 등록된 데이터 요소는 Control Committee의 데이터요소 투표를 통해 수용여부가 결정된다. 최초 등록된 데이터요소는 등록상태가 'Recorded'로 되며 데이터요소 투표가 부결되면 MDR 시스템으로부터 삭제되는 'DELETE' 연산이 발생한다. 표 3은 데이터요소의 삽입, 수정, 삭제 연산과 제약조건을 요약한 것이다.

**4. SQL/MDR 정의**

이 장에서는 ISO/IEC 11179의 명세를 바탕으로 MDR 구성요소들의 표준 인터페이스를 정의하고 3장에서 분석한 메타데이터 레지스트리 기본연산과 연산 패턴을 기반으로 검색연산에 대한 MDR 연산자를 정의한다.

**4.1 검색 연산자**

메타데이터 레지스트리 관리 및 접근을 위한 연산들은 일정한 패턴을 지니며 이미 3장에서 이러한 연산 패턴을 분석한 내용을 기술하였다. 분석한 연산 패턴들을 기반으로 하여 일관성 있는 메타데이터 레지스트리 접근방법인 SQL/MDR을 설계하기 위해서는 각 연산 패턴들에 대한 메타데이터 레지스트리 연산자를 정의해야 한다. 그림 5는 앞서 추출한 연산 패턴을 크게 네 가지로 분류하였고 각 분류에 대한 추상연산자를 보여주고 있다.

정의된 연산자는 ISO/IEC 11179의 명세를 준수하면서 기존에 구축되어 운영되고 있는 실제 MDR 시스템

표 2 검색연산

구분	검색 연산	옵션
데이터요소 검색	데이터요소 정의별 검색	<ul style="list-style-type: none"> <li>• KEYWORD와 일치, 시작 또는 포함된 그룹핑요소 검색 옵션 : Exact, Beginning with, Contained</li> </ul>
	데이터요소 이름별 검색	
	데이터요소 클래스별 검색	
	데이터요소 개념별 검색	
	데이터요소 개념영역별 검색	
그룹핑요소 검색	데이터요소 등록상태별 검색	• Candidate/Recorded/Qualified 등
	객체 클래스 검색	<ul style="list-style-type: none"> <li>• KEYWORD와 일치, 시작 또는 포함된 그룹핑요소 검색 옵션 : Exact, Beginning with, Contained</li> </ul>
	개념 검색	
	개념영역 검색	

표 3 데이터요소의 삽입, 수정, 삭제 연산

연산	설명	제약조건
삽입	데이터요소 신규등록	데이터요소 이름 중복 검사, 필수항목 누락여부 검사 및 메타정보 데이터타입 검사
수정	데이터요소 상태변경 및 개정	상태변경 및 개정에 따른 비전관련 정보 갱신
삭제	Recorded 데이터요소 투표부결	투표결과 체크 후 부결된 데이터 요소 삭제

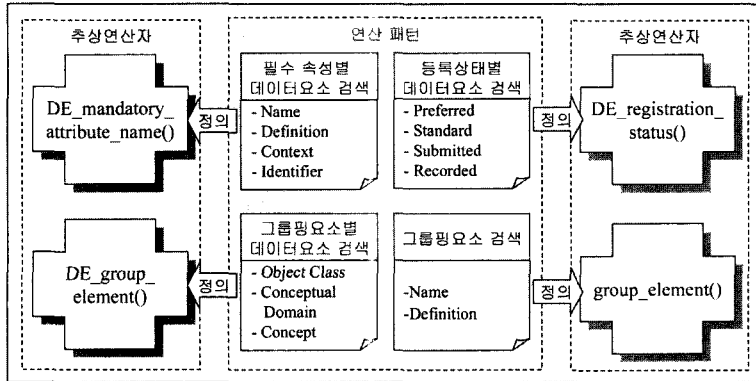


그림 5 검색패턴별 연산자 정의

표 4 연산자 분류

연산자	설명
DE_mandatory_attribute_name('KEYWORD')	필수 속성별 KEYWORD 매칭에 의한 데이터요소 검색
DE_registration_status('KEYWORD')	등록상태별 KEYWORD 매칭에 의한 데이터요소 검색
DE_group_element('KEYWORD')	그룹핑요소별 KEYWORD 매칭에 의한 데이터요소 검색
group_element('KEYWORD')	KEYWORD 매칭에 의한 그룹핑요소 검색

표 5 DE\_mandatory\_attribute\_name() 세부 연산자

연산자	설명
DE_name('KEYWORD')	데이터요소 이름이 KEYWORD인 데이터요소를 검색
DE_definition('KEYWORD')	데이터요소 정의가 KEYWORD인 데이터요소를 검색
DE_context('KEYWORD')	데이터요소 문맥이 KEYWORD인 데이터요소를 검색
DE_identifier('KEYWORD')	데이터요소 ID가 KEYWORD인 데이터요소를 검색

표 6 DE\_group\_element() 세부 연산자

연산자	설명
DE_object_class('KEYWORD')	데이터요소 클래스가 KEYWORD인 데이터요소를 검색
DE_conceptual_domain('KEYWORD')	개념 영역이 KEYWORD인 데이터요소를 검색
DE_concept('KEYWORD')	데이터요소 개념이 KEYWORD인 데이터요소를 검색

의 분석을 통해 기존 MDR 시스템까지 수용할 수 있도록 가능한 모든 연산자를 수용하도록 정의 하였다.

표 4는 검색시 사용되는 메타데이터 레지스트리 연산자를 특성별로 분류한 추상 연산자들을 보여준다.

추상 연산자 DE\_mandatory\_attribute\_name()은 데이터요소의 필수속성을 검색 키워드로 하여 검색하는 연산자이고 추상 연산자 DE\_registration\_status()은 등록상태별로 데이터요소를 검색하는 연산자이다. 추상 연산자 DE\_group\_element()은 그룹핑요소 이름을 검색 키워드로 하여 그룹핑요소별로 데이터요소를 검색하며, 추상 연산자 group\_element()은 그룹핑요소의 이름을 검색 키워드로 하여 특정 그룹핑요소를 검색한다. 분류 및 정의된 추상 연산자는 검색속성에 따라 여러 형태의 세부 연산자로 구체화된다.

표 5는 데이터요소의 필수 속성을 이용해 데이터요소를 검색하는 추상 연산자 DE\_mandatory\_attribute\_

name()의 세부 연산자를 정의한 것이다. 추상 연산자 DE\_mandatory\_attribute\_name()은 ISO/IEC 11179에서 명세하고 있는 데이터요소의 필수 속성을 이용하여 다양한 연산을 수행하는 세부연산을 갖는다. 이 연산자를 이용하여 사용자는 검색하고자 하는 데이터요소를 다양한 검색조건을 이용해 검색한다.

표 6은 그룹핑요소별로 데이터요소를 검색할 수 있는 추상 연산자 DE\_group\_element()의 세부 연산자를 정의한 것이다. 추상 연산자 DE\_group\_element()은 데이터요소를 검색할 때 그룹핑요소를 키워드로 사용하여 검색하는 세부 연산자로 구성되어 있다. 이들 연산자를 이용해 데이터요소를 그룹핑된 형태로 검색할 수 있으며 검색된 데이터요소를 통해 해당 그룹핑요소에 속하는 데이터요소를 확인할 수 있다.

표 7은 그룹핑요소 자체를 검색하는 추상 연산자 group\_element()의 세부 연산자를 정의한 것이다. 추상

표 7 group\_element() 세부 연산자

연산자	설명
object_class('KEYWORD')	객체클래스 이름이 KEYWORD인 객체클래스를 검색
conceptual_domain('KEYWORD')	개념영역 이름이 KEYWORD인 개념영역을 검색
concept('KEYWORD')	개념 이름이 KEYWORD인 개념을 검색

표 8 DE\_registration\_status() 세부 연산자

연산자	설명
DE_status(RA,DA,'KEYWORD')	등록상태가 RA이고 DA가 KEYWORD인 데이터요소를 검색 RA: Registration Attribute, DA: Data element Attribute
DE_status(RA)	등록상태가 RA인 DE를 검색
DE_status_submitted(DA,'KEYWORD')	등록상태가 submitted이고 DA가 KEYWORD인 DE를 검색
DE_status_recorded(DA,'KEYWORD')	등록상태가 recorded이고 DA가 KEYWORD인 DE를 검색
DE_status_submitted()	등록상태가 submitted인 DE를 검색
DE_status_recorded()	등록상태가 recorded인 DE를 검색

연산자 group\_element()은 그룹필요소 검색으로 그룹필요소인 객체클래스, 개념영역, 개념을 검색하는 연산자로 구성되어 있다. 각 연산자를 이용해 데이터요소를 논리적 그룹으로 분류하는 그룹필요소를 검색할 수 있다.

표 8은 등록 상태별로 데이터요소를 검색하는 추상 연산자 DE\_registration\_status()의 세부 연산자를 정의한 것이다. 추상 연산자 DE\_registration\_status()은 6개의 등록상태인 'Candidate', 'Recorded', 'Qualified', 'Standard', 'Preferred' 및 'Retired'별로 데이터요소를 검색할 수 있다. 이 연산자를 이용해 등록상태별 데이터요소의 검색이 가능하다.

4.2 MDR 구성요소의 표준 인터페이스 정의

이 절에서는 MDR 구성요소에 대한 표준 인터페이스를 정의한다. 여기서 표준이란 현재 인터페이스가 표준으로 사용되고 있다는 것을 뜻하는 것이 아닌 추후 SQL/MDR이 표준으로 사용되어야 한다는 것이며 이렇게 표준으로 될 경우 많은 장점을 갖게 된다.

ISO/IEC 11179에는 데이터요소를 위한 프레임워크를 명세하고 있을 뿐 MDR 필수 구성요소들에 대한 구체적인 명명법을 규정하지 않고 있으며 개발되는 MDR 시스템들은 서로 다른 구조를 지닌다. 이는 표준화된 MDR 구축을 어렵게 하고 MDR간 상호운용성을 저하시킨다. 이 논문에서는 ISO/IEC 11179의 명세를 바탕으로 필수 구성요소에 대한 인터페이스를 정의함으로써

논문에서 제시하는 표준 SQL/MDR의 유용성을 높이고자 한다. 표 9는 ISO/IEC 11179에서 명세하고 있는 필수 구성요소들에 대한 인터페이스 명을 보여준다.

MDR 구조 정의를 위해 표 9에서 정의한 인터페이스 명을 사용함으로써 각각 다른 구조를 갖는 MDR 시스템간 표준 인터페이스를 제공할 수 있다. MDR 구조정의를 위해 필요한 여러 상위요소 중 데이터요소와 객체클래스에 대한 상세 속성을 정의한다. 표 10은 ISO/IEC 11179에서 명세하고 있는 중요도가 높은 테이블인 데이터요소 속성들에 대한 인터페이스 명을 보여준다. MDR 시스템은 데이터요소 관리를 위해 데이터요소를 중심으로 운영된다.

MDR 시스템에 등록되어 있는 데이터요소는 다수의 속성들로 구성된다. 표 10에 정의된 속성은 데이터요소의 메타정보를 관리하기 위한 필수 속성으로 MDR 시스템은 이 속성 관리를 중심으로 여러 형태의 연산이 발생한다.

표 11은 객체 클래스에 대한 표준 인터페이스를 정의한 것으로 많은 데이터요소를 객체 클래스로 그룹화 한다. 표 10, 11과 같이 data\_element와 object\_class의 속성 인터페이스를 표준으로 정의함으로써 SQL/MDR을 보다 효과적으로 이용할 수 있도록 하고 ISO/IEC 11179 명세를 준수하도록 유도하여 MDR 시스템 간 상호운용성을 향상시킨다.

표 9 MDR 상위 구성요소를 위한 인터페이스 명

구분	상위 구성요소	설명
데이터 요소	DATA_ELEMENT	데이터요소 관리를 위한 테이블
객체 클래스	OBJECT_CLASS	객체 클래스 관리를 위한 테이블
개념 영역	CONCEPTUAL_DOMAIN	개념영역 관리를 위한 테이블
데이터요소개념	DATA_ELEMENT_CONCEPT	데이터요소 개념 관리를 위한 테이블
표현클래스	REPRESENTATION_CLASS	데이터요소의 표현클래스 관리를 위한 테이블
값 영역	VALUE_DOMAIN	값영역을 관리하기 위한 테이블



표 10 data\_element 속성 인터페이스 명

DE 속성 인터페이스 명	설 명
name	데이터요소 이름으로 중복되면 안된다.
identifier	데이터요소 식별자
version	데이터요소 버전으로 개정시 갱신된다.
registration_authority	데이터요소를 등록하는 기관
context	데이터요소의 문맥
definition	데이터요소의 상세정의
representation_category	데이터요소를 표현하는데 사용되는 기호
form_of_representation	데이터요소를 표현하는 형식
data_type	데이터요소의 데이터 타입
maximum_size	데이터요소 값을 저장하는 최대크기
minimum_size	데이터요소 값을 저장하는 최소크기
permissible_data_element_values	데이터요소의 허용가능한 값의 집합
responsible_organization	데이터요소의 책임 조직
registration_status	데이터요소가 등록된 상태
submitting_organization	데이터요소를 제출 조직

표 11 object\_class 속성 인터페이스 명

object_class 속성 인터페이스 명	설 명
identifier	객체 클래스 식별자
name	객체 클래스 이름
registration_status	등록상태
creation_date	객체 클래스 레코드 생성일
definition	객체클래스 정의
context	문맥

### 4.3 SQL/MDR 구문

앞서 분석한 메타데이터 레지스트리에서 사용되는 연산 패턴 중 검색유형에 대해 표준 SQL을 확장하여 BNF(Backus Naur Form)로 정의하였다. 검색패턴은

메타데이터 레지스트리에서 데이터요소를 검색하는 구문으로 표준 SQL문의 SELECT, FROM, WHERE 절에 MDR 연산자들을 추가, 정의하였다. 다음은 MDR을 위해 정의된 연산자들을 추가한 SQL/MDR의 BNF이다.

```

<extended query specification> ::= SELECT <extended attribute list>
                                   FROM <extended relation list>
                                   WHERE <extended attribute qualification>;
<extended attribute list> ::= <attribute list>|<MDR attribute list>|<MDR operator>;
<extended relation list> ::=
    [COMMA]<general relation list>|<MDR relation list> [<extend attribute list>];
<extended attribute qualification> ::= <general qualification>|<MDR qualification>;
<MDR qualification> ::= [<boolean term>]<MDR operator>[<extend attribute qualification>];
<MDR operator> ::=
    <DE mandatory attribute name> | <DE group element> | <group element>
    | <DE registration status> | DE_STATUS L_PAREN<MDR param list>R_PAREN;
<DE mandatory attribute name> ::=
    DE_NAME | DE_identifier | DE_DEFINITION | ... | DE_submitting_organization;
<DE group element> ::= DE_object_class | DE_conceptual_domain | DE_concept;
<group element> ::= object_class | conceptua_domain | concept;
<DE registration status>::=
    DE_STATUS_CANDIDATE | DE_STATUS_RECORDED | DE_STATUS_QUALIFIED
    | DE_STATUS_STANDARD | DE_STATUS_PREFERRED | DE_STATUS_RETIRED;
<MDR relation list> ::= DATA_ELEMENT | DATA_ELEMENT_CONCEPT | ... | OBJECT_CLASS;
    
```

표 12 data\_element 테이블

No	DE Name	DE Definition	Data type(Max)	Reg Status
1	Record_Reference_Number	문헌정보의 식별자	문자열(30)	Standard
2	Title_Korean	문헌정보의 한글제목	문자열(50)	Standard
3	Demand_Symbol	영역별로 분류하여 부여한 일련번호	문자열(10)	Standard
4	Research_Subject_Korean	한글로 표현된 연구과제명	문자열(50)	Standard
5	Research_Subject_English	영문으로 표현된 연구과제명	문자열(50)	Standard
6	Registration_Number	논문이 등록 된 번호	문자열(30)	Standard
7	Person_Name	문헌정보의 저자나 연구과제 책임자의 이름	문자열(20)	Standard
8	Coauthor_Name	공저자나 연구과제의 연구원의 이름	문자열(50)	Standard
9	Author_Index	저자나 책임자의 색인을 위한 키워드	문자열(20)	Standard
10	Academic_Degree	저자나 책임자의 학위	문자열(10)	Standard

4.4 SQL/MDR 질의 예제

이 절에서는 특정 질의문에 대해 표준 SQL문을 이용한 질의와 이미 정의된 SQL/MDR을 이용한 검색질의를 비교한다. 또한 이 논문에서 이미 정의한 표준 인터페이스인 data\_element와 object\_class의 속성 명을 사용한다. 표 12는 질의에 사용한 데이터요소 관리 테이블의 구조의 예이다.

[질의 1] 데이터요소의 이름이 'korea'인 요소를 검색하시오.

```
<SQL>
SELECT * FROM data_element
WHERE name = 'korea'
```

```
<SQL/MDR 1>
SELECT * FROM data_element
WHERE DE_name('korea')
```

```
<SQL/MDR 2>
SELECT DE_name('korea') FROM data_element
```

질의 1은 데이터요소를 이름별로 검색하는 예이다. 데이터요소 검색시 사용되는 연산자는 DE\_name()이며 질의 1과 같이 2개의 SQL/MDR문으로 각각 질의가 가능하다. SQL/MDR은 기존 SQL문과 비슷하며 연산자 DE\_name()에서 볼 수 있듯이 연산자 이름 자체에 데이터요소 이름검색이라는 의미를 포함하고 있어 사용자가 직관적으로 이해하여 손쉽게 사용할 수 있다.

[질의 2] 데이터요소의 이름이 'kor'으로 시작하는 요소를 검색하시오.

```
<SQL>
SELECT * FROM data_element
WHERE name LIKE 'kor%'
```

```
<SQL/MDR 1>
SELECT * FROM data_element
WHERE DE_name('kor%')
```

```
<SQL/MDR 2>
SELECT DE_name('kor%') FROM data_element
```

질의 2는 질의 1과 달리 데이터요소검색을 위해 기존 SQL문의 LIKE구문 용법과 유사한 형태를 적용한 예이다. 표준 SQL문의 LIKE문을 DE\_name('kor%') 형식으로 유사하게 이용함으로써 사용자가 좀더 손쉽게 응용할 수 있다.

[질의 3] 등록상태가 'submitted'인 데이터요소를 검색하시오.

```
<SQL>
SELECT * FROM data_element
WHERE registration_status = 'submitted'
```

```
<SQL/MDR 1>
SELECT * FROM data_element
WHERE DE_status_submitted()
```

```
<SQL/MDR 2>
SELECT DE_status_submitted() FROM data_element
```

```
<SQL/MDR 3>
SELECT * FROM data_element
WHERE DE_status(SUBMITTED)
```

질의 3은 등록상태별 데이터 요소검색 중 등록상태가 SUBMITTED인 데이터요소에 대한 검색 예이다. 질의 3은 4개의 SQL/MDR 형태로 다양한 질의가 가능하다.

[질의 4] 등록상태가 'submitted'이고 데이터요소의 이름이 'korea'인 데이터요소를 검색하시오.

```
<SQL>
SELECT * FROM data_element
WHERE registration_status = 'submitted' AND name = 'korea'
```

```
<SQL/MDR 1>
SELECT * FROM data_element
WHERE DE_status_submitted(name, 'korea')
```

<SQL/MDR 2>  
 SELECT DE\_status\_submitted(name,'korea') FROM data\_element

<SQL/MDR 3>  
 SELECT \* FROM data\_element  
 WHERE DE\_status(SUBMITTED,name,'korea')

<SQL/MDR 4>  
 SELECT DE\_status(SUBMITTED,name,'korea') FROM data\_element

<SQL/MDR 5>  
 SELECT \* FROM data\_element  
 WHERE DE\_status\_submitted() AND name='korea'

<SQL/MDR 6>  
 SELECT \* FROM data\_element  
 WHERE DE\_status(SUBMITTED) AND name='korea'

질의 4는 WHERE절에서 일반 질의와 표준 SQL 문법의 질의를 동시에 사용한 예이다. SQL/MDR은 질의 4와 같이 다양한 형태로 표현이 가능하며 일반적으로 사용하는 SQL 문법을 쉽게 적용할 수 있음을 보여주고 있다.

이처럼 SQL/MDR은 연산자 이름 자체에 MDR 시스템의 기능을 의미적으로 포함하고 있어 사용자가 손쉽게 사용할 수 있으며 기존 SQL 문법과 유사하기 때문에 자유롭게 다양한 문법 표현이 가능하다.

### 5. 구현 및 비교평가

이 장에서는 SQL/MDR을 위한 질의처리를 설계하고 구현한 결과에 대하여 기술한다. 이 논문의 초점은 SQL/MDR을 정의하는데 있다. 따라서 SQL/MDR 질의처리는 구현의 용이성을 위해 기존 관계형 데이터베이스 관리 시스템에 Wrapper를 탑재한 구조로 구현되었다.

#### 5.1 SQL/MDR 질의처리 구조

그림 6은 SQL/MDR 질의 처리기의 구조이다. SQL/MDR 구분분석기(Syntax Analyzer)와 SQL/MDR 의미분석기(Semantic Analyzer)는 사용자에게 의해 생성된 SQL/MDR 질의에 대한 구문 분석과 의미분석을 수행한다. 이를 위해 ANTLR(ANother Tool for Language Recognition)을 이용하여 구문 및 의미를 분석하는 자바코드를 생성하였다[15,16].

코드 생성기(Code Generator)는 질의 최적화에 의해 생성된 수행 순서에 따라 실행 트리를 생성하며 연산 처리기(Operation Processor)는 SQL/MDR과 일반 질의를 각각 분리하는 기능을 한다. 이렇게 생성된 질의

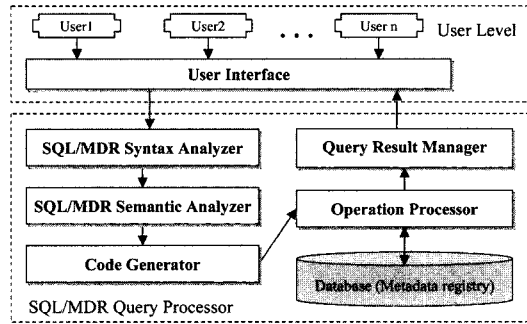


그림 6 SQL/MDR 질의처리기 구조

를 통해 질의 결과 관리자(Query Result Manager)는 질의 결과를 사용자에게 보여준다.

SQL/MDR을 처리하기 위해서는 SQL/MDR과 일반 SQL을 구분하여 질의 결과를 처리해야 한다. 그림 7은 이를 처리하기 위해 MDR 시스템 내의 SQL/MDR의 전체 흐름을 나타낸다.

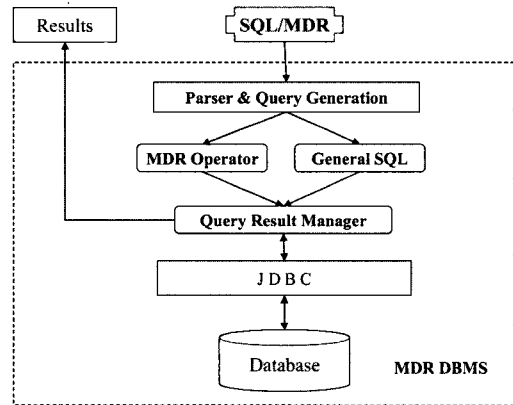


그림 7 SQL/MDR 질의 처리 전체 흐름도

SQL/MDR이 입력되면 MDR 시스템은 SQL/MDR을 파싱하여 MDR Operator와 General SQL의 Query로 분리한다. MDR Operator는 General SQL 형태로 생성되어 JDBC(Java Database Connectivity)를 통해 데이터베이스에 접근하여 해당 질의에 대한 결과를 가져온다. Query Result Manager는 질의에 대한 데이터베이스 응답을 받아 질의결과를 사용자에게 출력한다. 이런 전체 흐름을 따라 SQL/MDR은 분석되어 처리된다.

#### 5.2 구현

이 절에서는 실제 질의를 통해 5.1절에서 언급한 SQL/MDR 실제 처리과정을 간략하게 보여준다. 구현에 사용되는 질의는 SQL/MDR 구분과 표준 SQL 구문을 함께 사용하는 질의로 되어 있으며 질의와 이에 대한

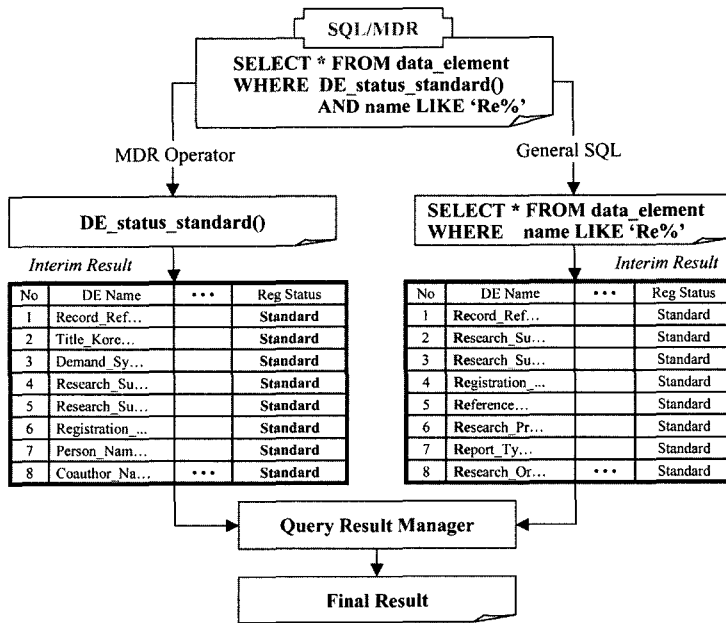


그림 8 SQL/MDR 질의문 처리 과정

No	DE Name	DE Definition	Value Type (Max)	Reg Status
1	Record_Reference_Number	문헌정보의 식별자로서 KISTI에서 부여하는 유일한 값을 가진다.	문자열(30)	Standard
2	Research_Subject_Korean	한글로표현된 연구과제명	문자열(60)	Standard
3	Research_Subject_English	영문으로표현된 연구과제명	문자열(60)	Standard
4	Registration_Number	논문이 등록된 번호	문자열(30)	Standard
5	Reference_Ebook_Title	논문이나 연구보고서에서 참조한 문헌명을 나타낸다.	문자열(300)	Standard
6	Research_Project_Subject	연구과제를 수행하는 부처에서 사용하는 사업명을 나타낸다.	문자열(30)	Standard
7	Report_yes	중간보고서 최종보고서 등과 같은 보고서의 유형을 나타낸다.	문자열(60)	Standard
8	Research_Organization_Name	연구과제를 수행하는 연구기관명을 나타낸다.	문자열(20)	Standard
9	Research_Organization_Code	연구과제를 수행하는 연구기관 코드를 나타낸다.	문자열(20)	Standard

9 개의 요소가 검색되었습니다.

그림 9 SQL/MDR 질의수행 결과

SQL/MDR은 다음과 같다.

[질의 5] 등록상태가 'Standard'이고 데이터요소이름이 'Re'로 시작되는 데이터요소를 검색하시오.

<MDR/SQL>

SELECT \* FROM data\_element

WHERE DE\_status\_standard() AND name LIKE 'Re%'

질의에 사용되는 SQL/MDR 연산자는 DE\_status\_standard()이고 LIKE를 이용한 SQL구문을 AND 연산과 동시에 사용하여 SQL/MDR 질의 처리기에 적용한

다. 그림 8은 SQL/MDR이 어떻게 MDR Operator와 General SQL로 분리 및 해석되어 처리되는지를 간략히 보여주고 있다.

그림 8과 같이 질의처리로 들어온 SQL/MDR은 Parser&Query Generator에 의해 MDR Operator와 General SQL로 분리된다. 각각 분리된 질의에 대한 수행결과는 그림 8과 같이 나타나며 그 결과가 Query Result Manager에 의해 Join되어 결과로 생성된다. 실제 생성된 구현결과는 그림 9와 같다.

5.3 비교평가

서지정보는 최근 국내에서 MDR 시스템에 대한 많은 연구가 진행되고 있는 분야로서 분산 관리되고 있는 서지정보 데이터베이스 간 데이터공유 및 통합검색을 위해 SQL/MDR을 적용하여 기존 접근방법과 비교하였다. SQL/MDR을 이용한 접근방법은 기존 방법에 비해 질의 모델링, 질의 처리 비용, 각 MDR 시스템 구조에 대한 독립성 유지 등 많은 장점을 지닌다. 이러한 장점들에 대하여 논의하기에 전에, 먼저 예제를 통해 기존 접근방법과 SQL/MDR 접근방법의 차이점에 대하여 기술한다.

그림 10은 두 접근방법을 비교하기 위해 가정한 두 개의 메타데이터 레지스트리 구조를 보여준다.

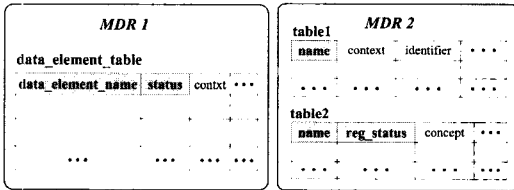


그림 10 MDR1과 MDR2의 구조

그림 10에서, MDR1은 테이블 data\_element\_table에 모든 데이터요소를 저장하며 MDR2는 두 개의 테이블, 즉 table1과 table2에 데이터요소가 분리하여 저장한다. 즉, MDR1과 MDR2는 서로 다른 구조를 지니며 각 테이블과 필드들 간의 의미 사상 관계는 다음과 같다.

- MDR1.data\_element\_table.status ↔ MDR2.table1.reg\_status.....①
- MDR1.data\_element\_table.data\_element\_name ↔ MDR2.table1.name.....②
- MDR2.table1.name ↔ MDR2.table2.name.....③

위의 사상관계는 두 필드간 의미 동등성을 표현한 것으로 서로 의미가 동일한 필드임을 의미한다. 위 표현들은 first.second.third로 구성되는데 first는 MDR을 의미하고 second는 테이블명을 그리고 third는 필드명을 나타낸다. 그러므로 ①의 경우에, MDR1내의 테이블

data\_element\_table에 있는 status 필드와 MDR2내에 있는 테이블인 table1의 reg\_status 필드는 상호 의미가 동일함을 나타낸다.

서로 다른 구조를 지니는 MDR1과 MDR2에 대해 질의 6이 주어질 경우, SQL/MDR 접근방식과 기존 접근방식에 의해 생성되는 질의문은 다음과 같다.

[질의 6] 메타데이터 레지스트리 MDR1과 MDR2로부터 등록 상태가 'RECORDED'인 데이터 요소 모두를 검색하시오.

```

<SQL/MDR>
SELECT DE_status(RECORDED)
FROM data_element.....<Q1>
<SQL_MDR1>
SELECT data_element_name
FROM data_element_table
WHERE status='RECORDED'.....<Q2>
<SQL_MDR2>
SELECT table1.name
FROM table1, table2
WHERE table1.name=table2.name
AND table2.reg_status='RECORDED'.....<Q3>
    
```

만약 MDR 시스템이 SQL/MDR을 기반으로 개발될 경우 질의문 <Q1>만을 이용하여 MDR1과 MDR2에 대한 검색이 가능하다. 하지만 기존 접근방법의 경우 MDR1과 MDR2가 서로 다른 구조를 지니고 있으며 일관성 있는 접근 인터페이스가 제공되지 않기 때문에 <Q2>, <Q3>과 같이 각각의 질의문을 작성하여 검색해야 한다. 또한 검색결과와 필드명이 다르기 때문에 최종결과를 얻기 위해 추가적인 통합과정과 이에 따른 비용이 요구된다.

이와 같이 다른 구조를 갖는 MDR 시스템이 N개 있을 경우, 기존 접근 방법을 이용하여 MDR 시스템에 접근하기 위해서는 N개의 질의문이 필요하다. 그러나 SQL/MDR 접근 방법의 경우에는 하나의 질의문을 통해 N개의 MDR로부터 결과를 얻을 수 있다.

표 13은 이에 대한 기존 접근방법과 SQL/MDR 접근 방법을 여러 가지 측면에서 비교한 결과이다.

접근방법의 표준화 측면에서, 기존 접근 방법은 MDR에 대한 특별한 접근방법을 제공하지 않는다. 그러므로

표 13 기존 접근방법과 SQL/MDR 접근방법 비교평가

비교항목	기존 접근방법	SQL/MDR 접근방법
접근방법의 표준화	미 지원	지 원
질의문의 간편성	복 잡	간 편
사용 용이성	어려움(다양한 접근방법 사용)	쉬움(단일 접근 방법)
질의모델링 비용	많음(질의문 제 작성)	적음(중립적인 모델링)
분산질의 처리 비용	많 음	적 음
MDR간 교환메커니즘	복 잡	단 순
MDR간 상호운용성	낮 음	높 음

사용자가 다른 MDR에 접근하고자 할 경우, 해당 MDR 시스템의 구조를 미리 파악하고 그에 알맞은 질의문을 작성해야 한다. 반면, SQL/MDR은 사용자가 MDR 시스템의 구조와 상관없이 표준화된 연산자를 이용하여 일관성 있게 MDR에 접근할 수 있다.

질의모델링 측면에서, 기존 접근 방법은 여러 MDR 시스템에 접근시 다양한 접근 방법을 사용함으로써 접근이 용이하지 않고 각 MDR의 구조를 사전에 파악하고 있어야 한다. 이는 질의문 작성을 어렵게 하고 각각의 MDR 시스템 구조에 적합한 질의문을 작성해야하므로 많은 질의모델링 비용을 요구한다. 그러나 SQL/MDR은 접근 방법이 표준화 되어 있어 여러 MDR 시스템에 대한 단일한 접근을 제공하기 때문에 사용이 용이하고 별도의 질의문을 작성할 필요가 없기 때문에 상대적으로 적은 질의모델링 비용이 요구된다.

기존 방법의 질의모델링 비용 문제는 분산되어 있는 MDR에 대한 분산질의 처리 비용 측면에서도 제한된 방법에 비해 많은 비용을 요구한다. 즉, 분산되어 있는 MDR의 구조에 대한 이해 및 질의 작성을 위해 많은 비용이 요구되며, 또한 분산질의를 통해 얻은 결과들의 이질성(예: 필드명의 불일치)으로 인한 결과 통합에 많은 비용이 요구된다. 그러나 SQL/MDR 접근방법의 경우, 하나의 질의문을 통해 분산질의가 가능하며 각 결과들의 구조 또한 일치하기 때문에 적은 분산질의 처리 비용이 요구된다. 이는 결국 분산질의 처리 메카니즘이 단순함을 의미한다.

앞서 언급한 접근 방법의 표준화, 질의모델링의 용이성 및 분산질의의 메카니즘의 단순성 등은 MDR간의 교환 메카니즘을 용이하게 하고 상호운용성을 보다 향상시키는 결과를 가져온다.

마지막으로, 기존 MDR 시스템들이 ISO/IEC 11179의 명세를 기반으로 개발되었음에도 불구하고 다양한 형태의 프로세스 및 연산을 갖는 MDR 시스템이 개발되었다. 이 때문에 기존 접근방법은 일관성 있는 프로세스와 연산을 갖는 표준 MDR 구축이 어려웠다. SQL/MDR 접근 방법은 ISO/IEC 11179에서 명세한 필수적인 기능을 표준 인터페이스로 정의함으로써 각 MDR 시스템마다 일관성 있는 접근연산을 사용할 수 있도록 한다. 따라서 ISO/IEC 11179 명세를 준수하도록 유도하는 효과를 얻을 수 있다.

## 6. 결론 및 향후연구

MDR은 메타데이터의 동적 관리 및 상호운용성 향상을 위해 개발된 국제표준인 ISO/IEC 11179의 핵심 요소로서 이를 기반으로 한 많은 메타데이터 레지스트리가 구축되었다. 하지만 ISO/IEC 11179가 개략적인 프

레이워크만을 제시하고 있으므로 구축된 MDR 시스템마다 각기 다른 연산과 스키마를 사용하고 있으며 이에 따라 몇 가지 문제가 발생한다. 즉, MDR 시스템마다 동일한 연산 패턴에 대해 다른 방법으로 개발됨으로써 개발시간과 비용의 낭비를 초래하고 표준 접근방법의 부재에 따른 MDR간의 이질성 문제와 개발된 MDR 시스템마다 다른 구조 및 명칭 사용으로 인한 국제표준의 혼란을 야기한다.

이 논문에서는 먼저 ISO/IEC 11179 명세 및 구현된 MDR 시스템을 분석하고 정형화된 패턴을 추출하여 MDR 시스템의 연산 패턴을 분석하였다. 분석된 접근연산 패턴을 바탕으로 메타데이터 레지스트리 연산자를 정의하였다. SQL에 통합, 확장하여 SQL/MDR을 정의하였다. 또한 ISO/IEC 11179에서는 MDR 필수 구성요소들에 대한 구체적인 명명법을 규정하지 않기 때문에 데이터요소를 중심으로 MDR 시스템 구성에 필수적인 테이블과 속성에 대한 표준 인터페이스 명을 정의하였다. 정의된 SQL/MDR과 MDR 구성요소의 속성명을 MDR 시스템의 표준 인터페이스로 이용함으로써 개발 시간 및 비용의 단축과 ISO/IEC 11179의 명세를 준수하도록 유도할 수 있다.

현재 정의된 SQL/MDR의 연산자들은 실제 개발된 MDR 시스템들에서 이용하고 있는 연산자들을 모두 수용한다. 그러나 향후 추가적인 연산 패턴을 발견하여 이를 현재의 SQL/MDR에 추가 반영하는 작업이 지속적으로 이루어져야 할 것이다. 또한 현재는 구현의 용이성을 위해 Wrapper를 이용하여 기존 데이터베이스 관리 시스템을 확장하여 SQL/MDR 질의처리를 구현하였다. 그러나 보다 나은 처리 성능을 위해서는 MDR 연산자들을 밀접합하여 SQL/MDR을 지원하는 데이터베이스 관리 시스템의 구현이 요구된다.

## 참 고 문 헌

- [1] ISO/IEC JTC 1/SC 32/WG 3, "ISO/IEC 9075, Database Language SQL3, Part1~10," 1999.
- [2] ISO/IEC JTC 1/SC 32/WG 2, "ISO/IEC 11179: Specification and standardization of data elements, Part 1~6," 2003.
- [3] ETRI, Research on the Registration and Search System of Component, Research Report, 2000.
- [4] KISTI, A study on the development of standardization and management model for science and technology information, Research Report, 2002.
- [5] EPA, Environmental Data Registry, <http://www.epa.gov/edr/>.
- [6] EPA, "Data Standards Publications and Guidelines," 2003.
- [7] AIHW, Australian National Health Information

Knowledgebase, <http://www.aihw.gov.au/>.

[8] Australian National Health Data Committee, "National Health Data Dictionary," 2003.

[9] U.S. Transportation System, <http://www.dot.gov/>.

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

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

[12] 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.

[13] 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.

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

[15] ANOther Tool for Language Recognition, <http://www.antlr.org/>.

[16] Terence Parr, "ANTLR Reference Manual," 2003.



박수현

1988년 고려대학교 컴퓨터학과 이학사  
 1990년 고려대학교 전산학 이학석사  
 1998년 고려대학교 컴퓨터학과 이학박사  
 1990년~1999년 (주)LG전자 중앙연구소  
 선임연구원. 1999년~2001년 동의대학교  
 공과대학 소프트웨어공학과 교수. 2002  
 년~현재 국민대학교 비즈니스IT학부 교수. 관심분야는 이  
 동통신 시스템, Active Network, 컴포넌트 기반 개발

백두권

정보과학회논문지 : 데이터베이스  
 제 31 권 제 3 호 참조



신동길

2000년 광운대학교 컴퓨터공학과(학사)  
 2002년~현재 고려대학교 대학원 컴퓨터  
 학과 석사과정. 관심분야는 이동 에이전  
 트, 데이터베이스, 메타데이터 레지스트  
 라



김영갑

2001년 고려대학교 식량자원학과 학사  
 (컴퓨터학과 부전공). 2003년 고려대학교  
 컴퓨터학과 석사. 2003년~현재 고려대학  
 교 컴퓨터학과 박사과정. 관심분야는 보  
 안 프로토콜, 이동코드 보안, 보안 명세

정동원

정보과학회논문지 : 데이터베이스  
 제 31 권 제 3 호 참조