SQLMM MDR: Query Language for Sharing and Exchanging of Metadata between MDRs

Sept. 6, 2008
Wuhan, China

Dongwon Jeong (djeong@kunsan.ac.kr), Kunsan National University
Yixin Jing (yixin_jing@korea.ac.kr)
Doo-Kwon Baik (baikdk@korea.ac.kr), Korea University
Contents

- Background and Need
- Possible Solutions
- Purpose of Our Proposal
- SQLMM MDR
  - How to Define MDR Operators
  - Query Patterns of Metadata Registry
  - Summary of MDR (Search) Query Patterns
  - An Example of Search Operators
  - A Brief BNF for Search Query
  - Examples of SQLMM MDR
  - Advantages
  - Restriction (Disadvantage? Barrier?)
- Conclusion
Background and Need

Development of MDR (ISO/IEC 11179)

Various MDRs have been built (EDR, ANHIK, etc.)

No method to make MDRs strictly following the standard (No mandatory attributes, Misuse)

Develop and use different access methods each other (Heterogeneity between MDR management systems)

Inconsistency Issue between MDRs (Incompatibility issue of metadata)

Difficult to realize an integrated metadata registry system (Integration of relevant and similar MDRs)

Need to get metadata from one or more MDRs (for building a database according to predefined DEs)

Need to harmonize various MDRs (MDR Integration: Tightly or Loosely Coupled)

How to resolve the issues?
Background and Need (cnt.)

Key Points to Resolve the Issues

- How to make valid metadata registries (following the standard, ISO/IEC 11179)
  - How to guarantee the validation of metadata registries

- How to consistently share and exchange metadata between MDRs
  - A consistent and standardized access method
Possible Solutions

- **Interface Mapping between MDR Management Systems**
  - Required for implementation of one-to-one adaptor
  - Do not guarantee metadata registries to be valid

- **Standardized Binding API-based Approach**
  - Every system should implement the binding API
  - For example, ISO/IEC 20944
  - Do not guarantee metadata registries to be valid

- **Direct DB Access**
  - Users should know the schema structures of the corresponding MDRs
  - Do not guarantee metadata registries to be valid

- **Standardized Query Language-based Approach**
  - DBMSs should support this language (Require an extension)
  - No need to implement any functions such as binding interfaces
    - DBMS developers are responsible for the implementation
  - Guarantee metadata registries to be valid
  - Our Approach
Purpose of Our Proposal

- **Goal of the Proposal**
  - To develop a SQL-based query language for consistent access (share and exchange) to metadata registries
  - This query language is named SQLMM MDR

- **How to achieve the goal**
  - Analyze metadata registry access patterns,
  - Define MDR operators, and
  - Integrate the operators into SQL (i.e., extend SQL)
How to Define MDR Operators

- Considerations to analyze operation patterns
  - Search Target: Data element vs. the others (i.e., data element concept, conceptual domain, value domain, object class, etc.)
  - Mandatory vs. Optional attributes
  - Life cycle of data elements: Submitted, Registered, Qualified, Standard, etc.
  - and ......
How to Define MDR Operators (cnt.)

- A Methodology to achieve the goal: Definition Processes

Analyzing query patterns

Components of 11179

Life Cycle of DE

Mandatory or Optional?

Defining operators

Search Operators

Update Operators (Add, Update, and Delete)

DDL (Create, Rename, Delete, etc.)

DCL

MRL

MML

MDL

MCL
Query Patterns of Metadata Registry

- **Key components of metadata registry**
- **Data element vs. Group element**
  - Most components are used as a logical grouping unit of data elements.
  - Group elements (GEs)
    - can logically cluster data elements
    - For example, conceptual domain, data element concept, class object, and so on.
  - Search targets are classified into two classes: DE and GEs
Query Patterns of Metadata Registry (cont.)

- **Lifecycle of a data element**
  - Submitted, Recorded, Qualified, Standard, Preferred standard, and Retired

  ![Lifecycle Diagram](image)

  **Notation**:
  - SER: Submitter
  - SUB: Submitted
  - REC: Recorded
  - QUA: Qualified
  - STA: Standard
  - PRE: Preferred standard
  - RET: Retired
  - HIS: Retired data element set

- **A Data element has mandatory attributes and optional attributes.**
  - **Mandatory attributes**
    - Must be captured to achieve the goal of this paper
    - For example, name, definition, and context of data elements should be described necessarily.
  - **Optional attributes**
    - There is no need to be captured indispensably, but ???
Summary of MDR (Search) Query Patterns

- MDR query operators for searching are defined by using the following factors:

  - Service patterns of the existing MDR management systems
  - Key objects of MDR
  - DE lifecycle (Registration statuses)
  - Mandatory DE attributes

  **Search Operator types extracted**
  - DE search operators with mandatory attributes
  - DE search operators with registration statuses
  - DE search operators using GE (Group elements)
  - GE search operators
  - Search options (exact matching, partial matching, ...)

  **MDR Search Operators**

  - MDR Operator types for DML
  - MDR Operator types for DDL

  **MDR Operators (for DML & DDL)**
User Interface

MDR Operator

```
select h.sub_emp.showName() from m_emp h;
```

Return de_name

MDR Operator(User-Defined Type)

MDL

MML

Query Executor

SQLMM MDR Layer

Metadata Layer

User Interface

User 1

User 2

User 3

User n-1

User n
The diagram illustrates the interaction between users and a DBMS (Database Management System) using SQL/MM MDR (User-Defined type Procedures And Methods). Users interact with the system through a user interface, which sends requests to the DBMS. The DBMS processes these requests using SQL/MM MDR and user-defined procedures.

Specifically, the diagram shows the following steps:

1. Users (User n, User n-1, etc.) send requests through the user interface.
2. The DBMS processes these requests using SQL/MM MDR and user-defined procedures, resulting in an interim result.
3. The interim result is then processed using general SQL, resulting in another interim result.
4. This final result is managed by the Query Result Manager, providing the final output to the users.

The diagram also includes a detailed table showing the DE_status_standard() function and its parameters, illustrating how the system processes data elements. The table includes columns for No, DE Name, and Reg Status, with specific examples of how data elements are processed.
An Example of Search Operators

**MDR Query Operators for Retrieval: Search Operators**

<table>
<thead>
<tr>
<th>Search Operators</th>
<th>Description</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE_name(KW, OPT)</td>
<td>Query operators to retrieve DEs with mandatory attributes</td>
<td>KW</td>
</tr>
<tr>
<td>DE_definition((KW, OPT)</td>
<td></td>
<td>OPT</td>
</tr>
<tr>
<td>DE_context(KW, OPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE_identifier(KW, OPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE_reg_organization(KW, OPT), ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE_status(RA, DA, KW, OPT)</td>
<td>Query operators to retrieve DEs with registration statuses</td>
<td>RA</td>
</tr>
<tr>
<td>DE_status(RA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE_status_submitted(DA, KW, OPT)</td>
<td></td>
<td>DA</td>
</tr>
<tr>
<td>DE_status_retired(DA, KW, OPT), ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE_object_class(KW, OPT)</td>
<td>Query operators to retrieve DEs using the group elements</td>
<td></td>
</tr>
<tr>
<td>DE_conceptual_domain(KW, OPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE_concept(KW, OPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>object_class(KW, PT)</td>
<td>Query operators to retrieve GE(Group Elements)</td>
<td></td>
</tr>
<tr>
<td>conceptual_domain(KW, OPT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>element_concept(KW, OPT), ...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Examples of SQLMM MDR

- **Query 1.** Retrieve all data elements where registration status is ‘RECORDED’.

  SQLMM MDR> SELECT DE_status.RECORDED()
  FROM data_element .................................................................(1-a)

- **Query 2.** List all data elements where registration status is 'RECORDED' and name is ‘KOREA’.

  SQLMM MDR> SELECT * FROM data_element
  WHERE DE_status.RECORDED()
  AND name='KOREA' ..............................................................(2-a)
Advantages

- Ease of use
- Familiarity to users
- **Consistent and Standardized access method**
  - Independent query description on physical structures (table structures)
  - Low modeling cost (decreases query description time)
  - Simplicity of query statement for accessing distributed registries
  - Low complexity for distributed query composition

**In its application aspect, SQLMM MDR**

- can be used Unified Access Method to Various Registries
- provides Unified query and Unified query result form
- can be used for developing distributed registries management system
- can be used as a communication protocol (method) for developing semi-automatic tools: Registries Mapping and Integrating tools
- Data Integration (among databases following their own registry)
Advantages: with a Scenario

- The below query shows advantages such as efficiency and simplicity of SQLMM MDR.
- Let’s assume the following situation to show contributions of SQLMM MDR.

Assumption (Situation): There are two registries with their own physical structure
- There exist two metadata registries.
- Each metadata registry has different MDR structure each other.
- The first metadata registry, MDR1 is designed as follows:
  1. data_element_table includes all of the data elements,
  2. data_element_name is a field name of data_element_table and means name of data elements, and
  3. Status is a field name of data_element_table and means registration status of data elements
- The second metadata registry, MDR2 consists of two tables as follows:
  1. table1 and table2 include all of the data elements together. Most attributes are included in table1 and some attributes including registration status are in table2,
  2. table1 and table2 use name as their join key, and
  3. Registration status is involved in reg_status, a field of table2.
Advantages: with a Scenario (cont.)

- Table Structures of Two MDRs, MDR1 and MDR2

<table>
<thead>
<tr>
<th>MDR1</th>
<th>MDR2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>data_element_table</code></td>
<td><code>DE_table1</code></td>
</tr>
<tr>
<td>*data_element_name</td>
<td>*name</td>
</tr>
<tr>
<td>definition</td>
<td>definition</td>
</tr>
<tr>
<td>status</td>
<td>status</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- Query 3. (Access to two metadata registries) Retrieve all data elements, where registration status is ‘RECORDED’ from metadata registries, MDR1 and MDR2.

\[
\begin{align*}
\text{SQLMM MDR} & \text{> SELECT } \text{DE\_status} \text{(RECORDED)} \\
& \text{FROM } \text{data\_element} \text{.........................(3-a)}
\end{align*}
\]

\[
\begin{align*}
\text{SQL} & \text{> SELECT } \text{data\_element\_name} \\
& \text{FROM } \text{data\_element\_table} \\
& \text{WHERE } \text{status} = \text{‘RECORDED’} \text{.................(3-b)}
\end{align*}
\]

\[
\begin{align*}
\text{SQL} & \text{> SELECT } \text{table1\_name} \\
& \text{FROM } \text{table1, table2} \\
& \text{WHERE } \text{table1\_name} = \text{table2\_name} \\
& \text{AND } \text{table2\_reg\_status} = \text{‘RECORDED’} \text{.........(3-c)}
\end{align*}
\]

- In case of SQLMM MDR based approach

- In case of use of SQL
  //query rewriting is required, i.e., requires several queries
  //high design time, complicated processing, etc.
Restriction (Disadvantage? Barrier?)

Query processor should know the followings:
1. $\text{data} \_\text{element} \Leftrightarrow \text{de} \_\text{table}$
2. $\text{name} \Leftrightarrow \text{de} \_\text{name}$
3. $\text{reg} \_\text{status}$ holds registration information

In a word, a **preprocessing** (mapping) is required for realization of the proposal.

1st approach to define SQLMM MDR: *Loose Coupling*  

Another approach to define SQLMM MDR: *Using a Predefined MDR Schema (Tightly Coupled)*

If a relational schema is given, this problem can be easily solved?!
Conclusion

- **SQL MM MDR**
  - SQL-based query language for consistent access to metadata registries (ISO/IEC 11179)
  - Advantages
    - Unified access method: One described, All returned (i.e., same query, same formatted results)
    - Strengthen standardization of registries