A Study on the Query Conversion System based on TMDR for Data Integration

Chigon Hwang[1], Seokjae Moon[1], Hyoyoung Shin[2], Gyedong Jung[1] and Youngkeun Choi[1]

[1] *Department of Computer Science and Engineering, Kwang-woon University*
447-1 Wolgye-dong, Nowon-gu, Seoul, 139-701, Korea
{duck1052, gdchung, choi}@kw.ac.kr
[2] *Department of Internet Information, Kyungbok College*
154 Sinpyeong-ro, Sinbuk-myeon, Pocheon-si, Gyeonggi-do, 487-717, Korea
hyshin@kyungbok.ac.kr

Abstract

This study suggests a query conversion method based on TMDR(Topic Maps MetaData Registry) which solves heterogeneity problem distributed in networks and integrates data efficiently. TMDR is a repository which combines Topic Maps with MSO(Meta Schema Ontology). TMDR provides an integrated knowledge for solving the problem of collision between schema and data. In order to integrate the distributed data, TMDR provides global schema and it solves heterogeneity problem within the local data using query conversion method. After analyzing relationship within MSO and Topic Maps, this method allows integrated access through ML(Meta Location) which manages local data.

The proposed method has several advantages. First, it can provide users who want to search unacquainted data with roadmap for intial data and schema. Second, the distribution for specific data can be checked. Third, it can find related information that is used for incorporating heterogeneous field. Fourth, the related problems such as homonym, heteronym, synonym for data can be solved using TMDR.

Keywords: *TMDR(Topic-Maps Meta Data Registry), data integration, Topic Maps, query conversion, Ontology*

1. Introduction

Most information resources are distributed among many systems, and such information needs to be integrated in response to various demands. Therefore, managing distributed data in a heterogenic computing environment is the most fundamental and difficult.

Information managed by distributed DB system has the problem of semantic and structural heterogeneity, and therefore, sharing resources for inter-operation may cause many problems. When sharing information real time, in particular, the information sharing system should have a capacity to detect and resolve the collision of semantic or logical structure. In most cases, however, information sharing needs to modify the existing legacy system. Therefore, data integration to guarantee the independence of individual legacy systems is needed, and there is a need to manage semantic collision of information. As a way of resolving such collision, the concept of ontology has emerged. There are many different definitions of ontology by field that is used as an integral part of an intelligent system. Gruber defined ontology as " formal and explicit specifications for shared conceptualization of the field." [1] As ontology has become a foundation of a knowledge-based system, the need of a language solely for ontology has been raised to correctly represent conceptualization structure, and standard languages such as RDF/RDFS[2], DAML[3], OWL[4], and Topic Maps[5] have been developed one by one. Among these, Topic Maps is a technical standard used to define the knowledge structure in a distributed environment and link this structure and knowledge resources, and can be said to be a new paradigm of the formation, extraction and navigation of information resources. XMDR is a system to integrate data by combining MDR and ontology to solve the problem of heterogeneity of distributed data and the technology to save XML-based relational DB meta-data in an object-oriented DB to resolve the heterogeneity of data as a result of data integration[6][7]. XMDR is, however, not enough to represent various kinds of ontology due to its lack of ontology representation and association. Therefore, we combine XMDR and Topic Maps as ontology, which is called TMDR, on the basis of which we propose a method of generating global query and converting it into local query.

In this study, chapter 2 examines MDR, XMDR, and Topic Maps as ontology, chapter 3 discusses rules and stages of TMDR-based query conversion method, chapter 4 looks into how query conversion affects the entire system, and finally chapter 5 concludes the study.

2. Related Studies

2.1. XMDR

For XMDR, standards should be set up by defining elements of DB of each node so as to prevent the heterogeneity of a system. Accordingly, XMDR secures the effectiveness of data by defining document structure. By making sure that change of schema in a node does not lead to random change of XMDR, credibility of data interchange should be secure.[8]

XMDR is the result of combining MSO to prevent the heterogeneity of meta-data and ML with location info of each node and information on access authority as well as InSO to avoid the heterogeneity of actual data values [6].

Although XMDR is useful in resolving the problem of heterogeneity in schema among local data to integrate data, it has a limit in resolving the semantic heterogeneity by means of ontology. Therefore, as a way of representing ontology, TMDR combining Topic Maps that emphasizes association is used.

2.2. Topic Maps

By adding topic/occurrence to the topic/association model, Topic Maps can function as a link between knowledge representation and information management. Knowledge is clearly different from information; knowing something is different from having information about it. In this respect, knowledge management is reduced to three activities - 'generation,' 'formalization,' and 'delivery.' Topic Maps is the standard for formalization among them and essential to develop a tool for generation and delivery[5][10].

Components of Topic Maps are as follows [10]:

- Topic Class: also known as Topic Type, which means the classification of topics.
- Topic: It represents subject. Generating Topic that represent a certain theme in computer means that Topic convert the theme into an object which can be understood and processed. Topic is instance of Topic Class.

- Association: used to set the relationship between Topics defined in Topic Maps and thereby provide context of Topic in Topic Maps. Setting the relationship between Topics is essential to model knowledge.
- Occurrence: a link of Topic-related information resources. When information resources provide information on Topic, it is represented as occurrence for the applicable Topic.

Among standards of Topic Maps are SGML-based ISO/IEC 13250:2000 and XTM 1.0 using XML syntactic system for the web environment [11].

Like RDF/S and OWL, XTM 1.0 provides a way of realizing Semantic Web and is used to represent information of relational DB. While DB represents the relationship between information objects only, XTM 1.0 can connect different locations where information objects exist[12]. However, most of the current tools to build Topic Maps are a stand-alone type that accesses an XTM and process it directly. This type can process XTM documents only, but it takes long to process XTM documents, and the type cannot process bulk Topic Maps data. To solve this problem, key technologies of Topic Maps based on XMDR are combined on the basis of TMDR.

3. TMDR-Based Query Conversion

3.1 System Design for Query Conversion

The proposed system consists of three layers like Fig. 1. There exists service process layer, data hub layer, and data layer. The service process layer performs the role of manager between user interface and lower layers. The data hub layer controls access of TMDR and manages integration of data. The data layer administrates TMDR and proxy. The proxy is used for saving collected data temporarily and selecting relation of data. The user interface and application programs are interconnected for user to access the system and the local systems store distributed data. This chapter will describe three layers and each layer's agent.

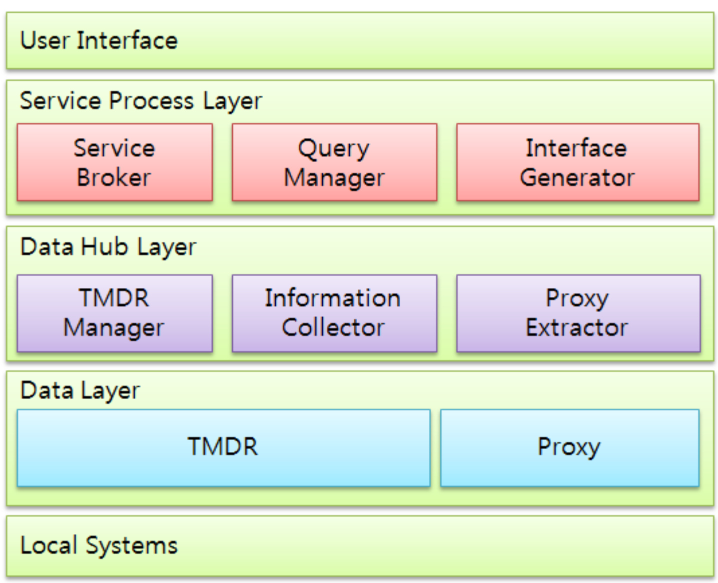

Fig. 1. The Query Conversion System based on TMDR for Data Integration

3.1.1 Service Process Layer

The service process layer provides user with service. This layer handles service interface, query conversion, information retrieval, generation of interface, and modification of expert network.

- Service Broker : jnterconnects and controls several agents for efficient service
- Query Manager : converts global format information into query format appropriate to locality
- Interface Generator : generates a format appropriate to interface through input keyword and category information

3.1.2 Data Hub Layer

Data hub layer provides buffer for integrating data. This layer manages TMDR and solves heterogeneity problems. It manages information that was collected from distributed source, and notifies to service process layer.

- TMDR Manager : TMDR Manager is a set of agents to manage TMDR. Each agents provide information for converting query in which it uses mapping information between standard item of whole system and local item, and provides information to resolve data heterogeneity.
- Information Collector : Information Collector collects the result of grid infrastructure and stores it to the proxy. The status of collection is notified to service broker. This data is transferred by XML format and stored to the proxy based on TMDR standard. It is offered to user interface through interface generator.
- Proxy Extractor : Proxy Extractor extracts required information for expert network after analyzing research finding.

3.1.3 Data Layer

Data layer takes a role of providing storage for the proposed system. This layer consists of TMDR and proxy.

- TMDR : TMDR consists of MDR and Topic Maps. MDR determines the standard of meta data and constructs mapping information by analyzing relationship with meta data of local system. MDR consists of MSO and ML. MSO stores meta data and mapping information for integrating legacy of data layer, ML has location and security information to manage the access of legacy. Topic Maps is a repository which has relationship with meta data and data which analyze their collision. Topic Maps consists of topic class, topic, and association. ML takes a role of Occurrence of Topic Maps, and this knowledge is used for integrating distributed data over the network.
- Proxy : Proxy stores the processing result of user request temporarily. This repository's structure depends on standard information of TMDR and creates tables automatically to perform process. Each name of table will use the process ID and time stamp.

3.2. TMDR for Generation and Conversion of Query

TMDR is a combination of MDR and Topic Maps of XMDR to integrate data. MDR consists of MSO and ML. MSO is a map of the semantic and structural relationship between global items and local ones. ML manages access information of local system and information for security management. The role of Occurrence of Topic Maps is possible by ML of MDR. Schema Topic used in Topic Maps becomes global schema of MDR. Other elements than Occurrence use the method proposed by Topic Maps.

Figure 2 shows how the search using TMDR works, which generates global query to find information on patients of "general surgery." Basic search condition is department = 'general surgery.' Here, "department" is global item used as Schema Topic, and ‘ general surgery' is Instance Topic. In regards to association of these Topics, there are synonyms 'gs' and ’ surgery' found, along with sub-keyword ‘ orthopedic.' These associations are included in global query according to the choice of the user, application task, and system configuration to generate global query. The process of it is explained in detail in 3.3.

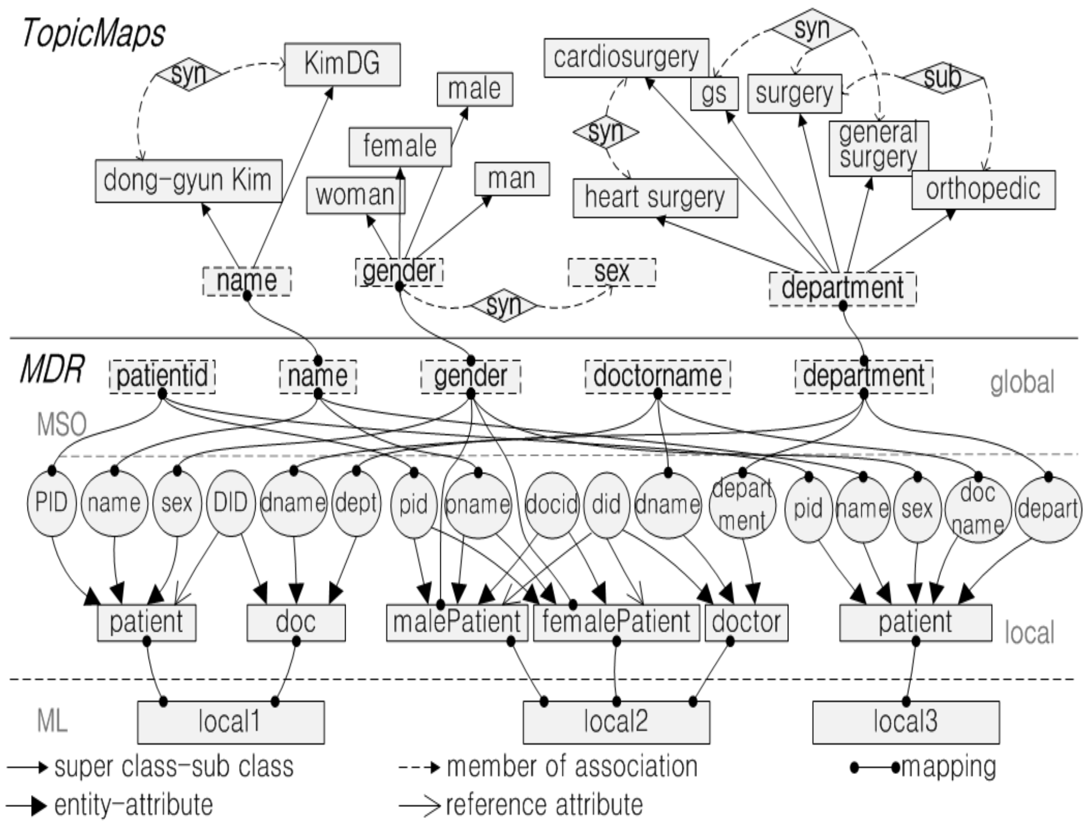

Fig. 2. Inter-operation between Topic Maps and XMDR at the TMDR.

TMDR-based data integration follows I=<G, L, M> like general data integration[13]. I is the result of the integration, which is done by global schema(G), local schema(L) and mapping of the two schemata(M). The result of the integrated queries is a global view, which consists of the union of local views. The set of tuple(t) written in a global view basically has the relationship of $t(Q^g) \sqsupseteq t(Q^l)$. In this relationship, Q^g means a global view, and Q^l refers to a local view, the result of search for local DB. The relationship between Q^g and Q^l is as follows:

$$q = q^{L_1D_1} \cup q^{L_2D_2} \cup q^{L_3D_3} \cup \dots \cup q^{L_nD_n} = \bigcup_{i=1}^{n} q^{L_iD_i}$$

Local query $q^{L_iD_i}$ is the result of the query that accesses local DB of the ith local system, and q is the union of $q^{L_iD_i}$.

There are six relationship sets composing TMDR. Since a relationship does not refer to individual tables, but it can consist of one or more tables, we call it a relationship group. This is the foundation domain(D) of data inter-operation, which consists of the relationship group D={S, L, M, T, A, m}. S, L, and M are relations included in MDR Area. They represent standard schema(S), local schema(L), and mapping of the two schemata(M), respectively. T and A is the relation included in Topic Maps. T is a set of Topic relation having Schema Topic and Instance Topic of Topic Maps. A is the relation set representing the association of Topic that has the relationship between Schema Topic and Topics. Finally, relation m is the relation where the mapping of MDR Area and Topic Maps Area in TMDR is stored.

3.3. TMDR-Based Global Query Generation

Users face many problems to obtain information. First, they need to know where information is and whether there is information they want. Second, they need to know access information of the system. Third, they need to know the method of query to access the system and extract information they look for. Fourth, they need schema information such as field name to carry out the query of the system. There are many issues other than these. These issues occur because the actual DB managed by the existing system is created by different schema. Authorized users in the integrated system should have access to data regardless of actual schema, and for this purpose, we use TMDR. This enables users to perform query by simply selecting information items and condition they want. The selected information serves as a resource to generate global query.

The format of global query is similar to SQL sentence except that it has no FROM clause to designate the table to search. Field name uses global item of MDR. Global item is a virtual schema for schemata of the existing local systems. A gs_name of this table is a global item. <global_item_list> in the format of global query is the list of gs_name. The basic format of global query is comprised of SELECT clause and WHERE clause to describe search condition as below.

SELECT <global_item_list> WHERE <condition>;

Here, <global_item_list> is a list of global items, which is provided to the user interface, and <condition> means search condition, which consists of global items in global query and condition value selected by the user. Global query is expanded by adding a semantic relation by means of the association of Topic Maps. In converting to local query, if there are two or more tables in legacy system, "join" condition is added to local query. When converted into local query, table info equivalent to FROM clause of SQL is automatically added by MDR. Table 1 shows the basic rules of SELECT clause.

Table 1. Rule of Global Query for SELECT Statement

```
SELECT <global_item_list> WHERE <condition>;

<global_item_list> ::= <gs_name> |
                       <global_item_list>, <gs_name>
<condition> ::= <comparison>|
                NOT(<condition>)|
                <condition>AND<comparison>|
                <condition>OR<comparison>
<comparison> ::= <gs_name><operation><gs_name>|
                 <gs_name><operation><value>|
                 <gs_name><operation><expr>
<operation> ::= = | != | > | >= | < | <= | LIKE
<value> ::= <number>|<string>
<expr> ::= <gs_name><arthmeric operation><value>
```

<global_item_list> means that a number of <gs_name> can be listed, and <gs_name> is global item described in the global_schema table. For <condition>, multiple conditions can be used through logic operator. <comparison> represents the format of conditional equation, where <expr> is a numerical equation, and <value> is a constant such as numbers or characters. If you want to search employees whose the actual pay (employee.takehome_pay) exceeds 1.2 times of the basic pay (employee.basic_pay), the condition is employee.takehome_pay > employee.basic_pay * 1.2, where employee.basic_pay * 1.2 is <expr>.

3.4. Expansion of Global Query and its Conversion into Local Query

Converting global query into local query takes eight steps, and processing local query takes two steps as in Figure 3, using TMDR of 3.1. These steps are as below:

Step 1. Create global query. At this step, to create global query, requests of the user based on global item provided by MSO of TMDR are created in the format of query in 1.

Step 2. Expand global query. At this step, by providing the user with the association available in the user interface by MSO and Topic Maps of TMDR, global queries are expanded by adding association via Topic Maps. Users can expand global query by adding necessary association in order to create more accurate and efficient query.

Step 3. Parse global queries. This step extracts <gs_name> and <value> from global queries to convert local query. <gs_name> is the name of standard item of a temporary table to gather results.

Step 4. Create a temporary table in Proxy to gather results. At this stage, temporary table is made to gather results in Proxy using <global_item_list> next to SELECT sentence and session information of the user.

Step 5. Extract TMDR. This step is subdivided into many sub-stages: First, local schema extraction via MSO from global item <gs_name> parsed at Step 3, second, extraction of <value> from associations of Topic Maps at Step 3, and third, extraction of legacy system

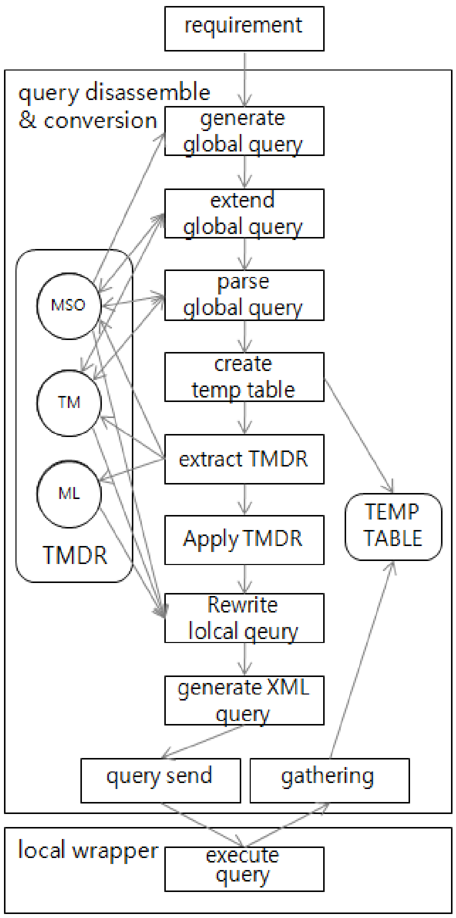

Fig. 3. Course of Query Conversion

access information of ML. By means of the relationship between global schema and local schema, access information of legacy system managed by ML is extracted.

Step 6. Apply TMDR. Global query is expanded by applying the association extracted at Step 5, and the association should be applied flexibly depending on the characteristics of tasks and DB. To search theses, it may be represented differently even if the key word is the same. In this case, the association should be added as "OR" in the search condition, but if it is divided into hospital or branch, every branch may have differently represented data. In this case, value substituted by branch should be applied.

Step 7. Reproduce local query. At this step, global item <gs_name> is mapped to local schema extracted at Step 5, and resulting difference in schema structure is resolved by means of combination, division, joint, and substitution of strings. At this step, mapped to local items, table info of legacy system is available. This table info is included into FROM clause of local query, completing local query.

Step 8. Create XML query. This step is designed to change local query converted to sent SOAP message to local system into XML message.

Step 9. Transmit query. Once created, local query is sent to Wrapper Agent to have access to local DB of each legacy system.

Step 10. Gather results. Wrapper Agent performs local query, of which result is returned by Gathering Agent. The returning result is inserted via an verification test to examine if it is fit to the table in Proxy made at Step 2 above, of which result is sent to the user by Gathering Agent notifying to the user interface.

For example, the above query is used for the query to search “ patients of general surgery. ” The schema structure to perform this query is as shown in Table 2. The picture depicts how global items seen to the user and local items participating in each example are mapped. The mapping structure may vary among legacy system. Local A is almost similar in structure, but consists of two tables - patient table and doctor table. Unlike Local A, Local B has a design of separated patient tables - male patient table and female patient table. Local C is the same as Global Schema in structure and detail, but not in schema representation.

Table. 2. Structure of Global and Local Schema

area	schema structures and associations					
global	attribute	patientid	name	gender	doctorname	department
	type	char(12)	char(20)	char(6)	char(20)	char(50)
local A	attribute	chartid	name	sex	did	
	type	char(12)	char(20)	char(6)	char(8)	
	attribute	did	doctor	dept		
	type	char(8)	char(20)	char(40)		
	association : generalsurgery = surgery(generalization)					
local B	attribute	ptid	pname	did		
	type	char(10)	char(20)	char(8)		
	attribute	ptid	pname	did		
	type	char(10)	char(20)	char(8)		
	attribute	did	dname	department		
	type	char(10)	char(20)	char(50)		
	association : generalsurgery = gs(acronym)					
local C	attribute	pid	name	sex	docname	depart
	type	char(12)	char(20)	char(6)	char(20)	char(50)
	association : generalsurgery = generalsurgery(equal)					

Global query created as in Table 2 is "SELECT patientid, name, gender, doctorname, department WHERE where department = "generalsurgery";" which is applied to Local B system in the above steps and results in local query as in Table 3.

Table 3. Applied Local Query for Local B.

```
SELECT a.ptid, a.pname, "male", c.dname, c.department
FROM localB.malepatient as a, localB.doctor as c
WHERE a.did = c.did
AND c.department="gs"
UNION
SELECT b.ptid, b.pname, "female", c.dnmae, c.department
FROM localB.femalepatient as b, localB.doctor as c
WHERE b.did = c.did
AND c.department="gs";
```

Query conversion like this generates local query by receiving query sentences of the previous step sent by the relevant Agent and performing tasks of each step.

4. Application and Performance Evaluation

4.1 Application

The proposed system processes the task into three classes. Fig. 4 represents the three classes into sequence diagram.

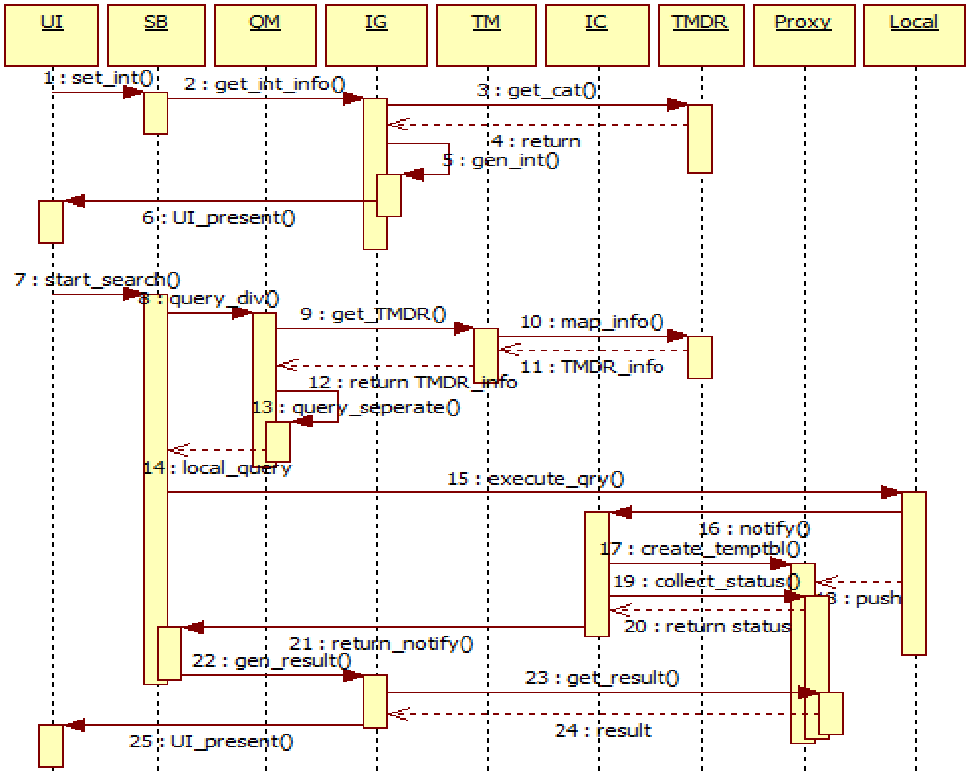

Fig. 4. The search process of expert using TMDR. UI:User Interface, SB:Service Broker, QM:Query Manager, IG:Interface Generator, TM:TMDR Manager, IC:Information Collector

First stage constructs input interface which provides category for user access and includes from number 1 to number 6. Table 4 describes detailed process used in first stage. Second stage divides query and adds link using search field of user and input condition. Standard item is converted to local item in this stage applying TMDR. These process includes from number 7 to number 14, and Table 5 describes the process of this stage. The divided queries are transmitted to each local system and the proxy collects results in third stage which includes number from 15 to number 25. Table 6 summarizes stage 3.

Table 4. The Process for Creating Interface

process name	descriptions
set_int()	create interface for user access
get_int_info()	request information to create interface
get_cat()	access to TMDR requested by IG
gen_int()	create interface
UI_present()	interface created in UI

Table 5. The Process of Dividing Query through Semantic Adjustment

process name	descriptions
start_search()	start search through search condition of user create the global query
query_div()	divide global query into local query
get_TMDR()	request mapping information of TMDR to divide query
map_info()	extract mapping information
query_seperate()	divide query

Table 6. The Process of Exchanging Data with Local System

process name	descriptions
execute_qry()	transmit the divided query to each local system
notify()	notify the result of processing
create_temptbl()	secure temporal storage in proxy
collect_status()	confirm the status of collection result
return_notify()	report the status of collection
gen_result()	request the creation of result to interface
get_result()	get the result of collection in proxy
UI_present()	display the result in UI

4.2 Performance Evaluation

In this section, we compare duration and actual query time of local systems by applying the proposed query conversion method. The result of the comparison is expressed in accumulated as in table 4.

Table 4. The Proportion of Conversion Time at Total Time

	query1	query2	query3	query4	query5	query6	query7
execute time	5,571ms	7,758ms	4,243ms	5,055ms	8,123ms	5,611ms	7,723ms
convert time	180ms	177ms	201ms	156ms	210ms	163ms	153ms
ratio	3.23%	2.28%	4.74%	3.09%	2.59%	2.91%	1.98%

Table 4 compares performance time and conversion time of different seven queries in each local system. The conversion time is the entire time included in the proposed system such as gathering time, as well as time spent to convert global query into local query. As in Table 4, query time varied by query characteristics of each local system and size of data. Also in the integrated system, there is almost no difference in time of this kind. Conversion time of the integrated system is trivial compared to the total time, which can be confirmed by the ratio of performance time and conversion time in Table 4. Therefore, if conversion time ratio has little significance, this system is seen as efficient because it can integrate data with global queries alone, not considering each local system.

5. Conclusion

As there have been many attempts to integrate data distributed in many locations in a network due to the characteristics of business or to disperse load with the existing system as it is, the study proposed a TMDR-based query conversion method for integration. We explained the rules of standard query sentences to convert queries and how they are converted into local queries via TMDR in these rules. Advantages of the proposed method are: First, it can be applied as a way of converting queries in various data integrations, and it is possible to access local systems simply by converting queries under the existing system. The proposed method not only applies to simple searches, but also enables an integrity test of data gathered by TMDR, since TMDR provides information on field suitability, structural form, and semantic relation. It is also efficient in gathering data for analysis like dataware house. The proposed method is expected to improve by continuously applying to queries for further studies.

References

[1] T.R.Gruber, "Towards principles for the design of ontologies used for knowledge sharing", International Journal of Human-Computer Studies, Vol.43, No.5, 1995, pp.1-2.

[2] http://www.w3.org/TR/rdf-schema/ W3C Recommendation 10 February 2004

[3] http://www.daml.org/ontologies, DAML.org Ontology Library. As of July, 25th 2003.
http://www.daml.org/ The DARPA Agent Markup Language (DAML)

[4] D. L. McGuinness and F. Harmelen. "OWL Web Ontology Language Overview," W3C Recommendation, 10 February 2004, http://www.w3.org/TR/owl-features/

[5] http://www.isotopicmaps.org/ ISO/IEC 13250, Topic Maps (Second Edition) 22 May 2002.

[6] SeokJae Moon, GyeDong Jung, YoungKeun Choi, "A Study on Cooperation System design for Business Process based on XMDR in Grid", International Journal of Grid and Distributed Computing Vol. 3, No. 3, September, 2010.

[7] Youn-Gyou Kook, Gye-Dong Jung and Young-Keun Choi, "Data Grid System Based on Agent for Interoperability of Distributed Data", Agent Computing and Multi-Agent Systems, PRIMA 2006, LNAI 4088, pp. 162–174, 2006.

[8] Kevin D. Keck and John L. McCarthy, "XMDR: Proposed Prototype Architecture Version 1.01", http://www.XMDR.org/, February 3, 2005.

[9] Ray Gates, "Introduction to MDR-Tutorial on ISO/IEC 11179“, Metadata Open Forum 2004, Xian, May 17, 2004.

[10] Steve Pepper, "The TAO of Topic Maps. In Proceedings of XML Europe 2000", Paris, France, 2000. http://www.ontopia.net/top-icmaps/materials/rdf.html

[11] Members of the Topicmap.org Authoring Group, "XML Topic Maps(XTM) 1.0", www.topicmaps.org/xtm/, 2001.8.

[12] Moore G. Ahmed K. Brodie A., "Topic Map Objects," TMRA, pp.166-174, 2007.

[13] M. Lenzerini. Data integration: A theoretical perspective. In Proceedings of the Symposium on Principles of Database Systems (PODS), pp 233-246, 2002.

Authors

Chigon Hwang received the B.S. degree in Business Administration from Changwon National University, Korea, in 1995. the M.S. degree in computer software from Kwangwoon University, Seoul, Korea, in 2004. Now, he is a Ph.D course in the Department of Computer Science and Engineering at Kwangwoon University, Seoul, Korea. His current research interests include eXtended metadata registry (XMDR),

distributed computing, grid computing, interoperability, semantic social network.

SeokJae Moon received the BS degree in computer science from Korea national open University, Seoul, Korea, in 2002, the MS degree in computer software from Kwangwoon University, Seoul, Korea, in 2004. Now, he is a PhD candidate in the Department of Computer Science and Engineering at Kwangwoon University, Seoul, Korea. His current research interests include eXtended metadata registry (XMDR), business process, distributed computing, grid computing, interoperability, cooperation systems.

HyoYoung Shin received the B.S. the M.S. and Ph.D degrees from the University of Kwangwoon, Seoul, Korea, in 1986, 1988, and 1998, all in computer science. Now, he is a Professor in the department of computer science , Kyungbok University, Pochun, Korea, His current research interests include metadata registry(MDR), mobile agents, distributed computing, computer network, grid computing, network security.

GyeDong Jung received the B.S. the M.S. and Ph.D degrees from the University of Kwangwoon, Seoul, Korea, in 1985, 1992, and 2000, all in computer science. **Now**, he is a Professor in the institute of Information and Science Education, Kwangwoon University, Seoul, Korea, His current research interests include metadata registry(MDR), mobile agents, web service, grid computing, XML, semantic social network.

YoungKeun Choi received the B.S. degree from in mathematics education from National University of Seoul, Seoul, Korea, in 1980, the M.S. and Ph.D degrees in computational statistics from Seoul National University, **Seoul**, Korea, in 1982 and 1989. Now, he is a Professor in the Department of Computer Science and Engineering at Kwangwoon University, His current research interests include object oriented design, mobile agents, interoperability, parallel processing, ontology.