韓國 컴퓨터情報學會 論文誌
第11卷 第3號, 2006. 7.

2006-11-3-1-11

계량화된 지식 추상화 계층을 이용한 협력적 질의 처리

신 명 근*

Cooperative Query Answering Using the Metricized Knowledge Abstraction Hierarchy

Shin, Myung Keun *

요 약

데이터베이스 시스템에서 협력적 질의응답이란 질의 내용과 정확히 일치하는 자료뿐만 아니라 좀 더 넓은 인근 범위의 자료 또는 근사적인 자료를 검색해 주는 것을 말한다. 협력적 질의응답은 질의 분석, 질의 유연화, 유연화된 질의에 따른 근사적 자료 제공의 세 단계로 이루어져 있다. 질의 유연화를 수행하기 위해 의미적 관계를 표현하는 지식추상화 방법과 자료 사이의 정량적인 유사도를 거리로 표현하는 방법들이 지식표현 방법으로 사용된다. 본 논문에서는 보다 효과적으로 질의 유연화 단계를 지원하기 위해, 다단계 데이터 추상화 계층과 거리 척도를 지원하는 계량화된 지식추상화 계층(MKAH: Metricized Knowledge Abstraction Hierarchy)을 제안 한다. MKAH는 카테고리화 될 수 있는 자료에 대해 질의 유연화를 효과적으로 지원하며 두 값 사이의 정량적인 의미상의 유사도를 제공하여, 질의 결과에 순위가 매겨질 수 있도록 한다. MKAH의 실용성과 효율성을 검증하기 위하여 경력직 검색 분야에 대한 원형 시스템을 구현해보았다. 다양한 실험을 통하여 MKAH가 풍부한 의미 표현이 가능하면서 질적으로도 높은 거리 척도를 제공해 준다는 것을 보였다. 그 결과 MKAH를 채택하는 도메인은 다른 정량적인 숫자 도메인과 호환될 수 있다는 점과, 큰 규모의 시스템을 만드는 데에도 장점이 있음을 확인하였다.

Abstract

Most conventional database systems support specific queries that are concerned only with data that match a query qualification precisely. A cooperative query answering supports query analysis, query relaxation and provides approximate answers as well as exact answers. The key problem in the cooperative answering is how to provide an approximate functionality for alphanumeric as well as categorical queries. In this paper, we propose a metricized knowledge abstraction

• 제1저자 : 신명근
• 접수일 : 2006.06.5, 심사일 : 2006.06.22, 심사완료일 : 2006.07.16
* (주)투이 컨설팅

hierarchy that supports multi-level data abstraction hierarchy and distance metric among data values. In order to facilitate the query relaxation, a knowledge representation framework has been adopted, which accommodates semantic relationships or distance metrics to represent similarities among data values. The numeric domains also compatibly incorporated in the knowledge abstraction hierarchy by calculating the distance between target record and neighbor records.

▸ Keyword : 협력적 질의처리(Cooperative Query Answering), 질의결과 순위결정(Ranking Query Results), 추상화 계층(Abstraction Hierarchy).

Ⅰ. Introduction

Conventional database systems usually get a query and generate the corresponding result. If the input query, however, has no answer, or the data is not available, or the query is not well-formed with respect to the database schema, a null answer or an error will be given. A cooperative query answering [2, 4, 5, 13, 14] can provide relevant information, permitting conceptual level queries of a wider scope or even approximate answers. To facilitate the query relaxation and the approximate information search, a knowledge representation framework has been one of the most important factors that have influence on performance the cooperative query answering system. A variety of studies on knowledge representation has been performed using logic models, semantic distance, abstraction, and so on.

The logic model approach [3, 8, 11] uses first-order logic predicates to represent the semantic relationship and integrity constraints among data values. Thus, the entire database consists of a set of base predicates. A database query is also written by a predicate rule whereby inquiredinformation is specified with free variables. The query is answered through conflict resolution and inference mechanism. However, this approach has limitations in guiding the query relaxation process and the less intuitive query answering process due to a lack of systematic organization. Therefore logic model approach is not adequate to build up a large scaled system.

The semantic distance approach [7, 12] uses the notion of semantic distance to represent the degree of similarity between data values. Every pair of data values within the data set is supposed to have semantic distances [12], and thus this approach provides straightforward and efficient method for query relaxation and provides ranked results sorted by the semantic distance. The ranked results help the user to find useful information among the results. For categorical data, the distance between two data values is stored in a table. However, since every pair is supposed to have semantic distances, the table size usually gets exploded when realistic application domain is considered, and thus as the table size gets larger, the maintenance cost becomes higher and the consistency of distance measure becomes harder to retain.

The abstraction approach uses the data abstraction method that has been considered as an effective method to accommodate the semantic relationships among data values [1]. In cooperative query answering, such data abstraction is also useful to associate data values for query relaxation. To validate the abstract concepts comprising the related data values, the type abstraction hierarchy [6] introduced the notions of subsumption, composition and abstraction and offered an integrated view of the type hierarchy with multi-level knowledge abstraction. The knowledge abstraction hierarchy (KAH) [10] extended the type abstraction hierarchy, and it focused capturing value abstraction information with additional abstraction information representing domain abstraction knowledge elicited from the underlying databases. However, these methods don't provide quantitative similarity measure among data values, which is a common limitation of the methods employing the abstraction approach. Hence, with the abstraction approach, the users could not decide the importance of the results. Also, since the abstraction approach does not match the system that uses semantic distance approach, this approach is not appropriate to build up a scalable and extendable cooperative query answering system.

In this paper, we propose a knowledge abstraction hierarchy that overcomes the problems discussed in the above. The hierarchy facilitates finding neighbor values for a target value in an easy way. We can obtain wider range neighbor values by choosing a higher-level abstract node of the target value. The distance information embedded in the hierarchy provides a more efficient method than the one based on a table. Since assigning distances are limited only to those small number of values directly linked to the new value, consistency in assigning distance data will be improved. It uses the hierarchy structure of abstraction approach and provides a quantitative measure between data values in the hierarchy.

II. Metricized Knowledge Abstraction Hierarchy

In this section we propose a metricized knowledge abstraction hierarchy that combines abstraction hierarchy with a semantic distance notion.

2.1 Abstraction Hierarchy

The MKAH is a knowledge representation framework that facilitates multilevel representation of data and meta-data for an underlying corporate database using data abstraction. Fig. 1 shows an instance of the MKAH that represents the abstraction information on Deformation. Values constituting the hierarchy may be parts of the underlying database or artificial values added to describe the semantic relationship among the existing data values. The MKAH is composed of two types of abstraction hierarchies: value abstraction hierarchy and domain abstraction hierarchy [10]. In the value abstraction hierarchy, there are abstraction relationships of specific node/abstract node. One node in a level can be generalized into an abstract node placed in an upper level. This abstraction relationship can be interpreted as IS-A relationship. For instance, `Rolling` is a (branch of) `Forming` while `Forming` is a (branch of) `Deformation`. As such, higher levels provide a more generalized data representation than lower ones and the root node can be interpreted as the most abstract, but representative name of the hierarchy. In Fig. 1, the root node is `Deformation` which can act as a representative of the hierarchy.

The leaf nodes are given with the level value 1, and the level value increases by one at a time when they are generalized with an abstract node. The *n level abstract node* of a specific node is the abstract node that is located in *n* level higher than the specific node. The path from a node to the root node is called an abstraction path, and the path from the root node to the node is called a specification path.

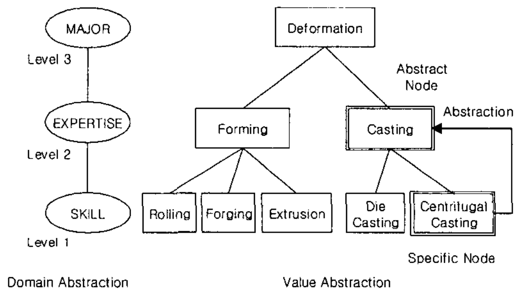


Fig. 1. Metricized knowledge abstraction hierarchy example on Deformation.

그림 1. Deformation에 대한 지식추상화 계층

Definition 1 (*Level difference*) The level difference value *n* between two arbitrary nodes is defined as the larger number of abstracted levels from the two nodes to their lowest common abstract node.

Definition 2 (*Neighbor nodes*) The *n* level neighbor nodes are the nodes that share a common abstract node in level difference *n*. (See Fig. 2)

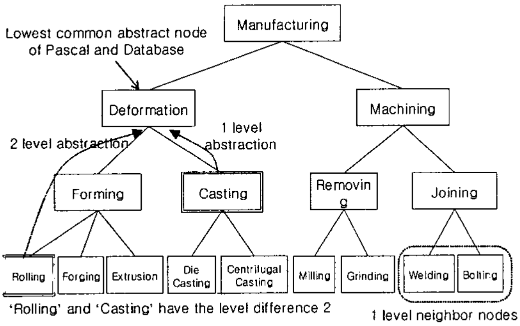


Fig. 2. Level difference and neighbor nodes.
그림 2. 레벨차이와 이웃노드

2.2 Definition of Distance and Its Calculation in the Hierarchy

In this section we represent the concept of the distance metric and definition of the shortest pass, and define the distance which satisfies the requirement of the distance metric. In adopting the distance metric into the MKAH, we should consider the following properties concerning the distances between nodes in the MKAH:

Property 1. *Distances between every pair of nodes can be derived.*

Property 2. *The smaller the level difference, the more similar the two nodes are semantically.*

Property 1 means that the distance between two arbitrary nodes in the MKAH should be provided. Property 2 means that the smaller the level difference is, the closer the distance should be. The smaller the distance, the more similar the two nodes are semantically. For example, the distance between `Forging` and `Extrusion` which have the level difference 1 should be closer than the distance between `Forging` and Die `Casting` which have the level difference 2.

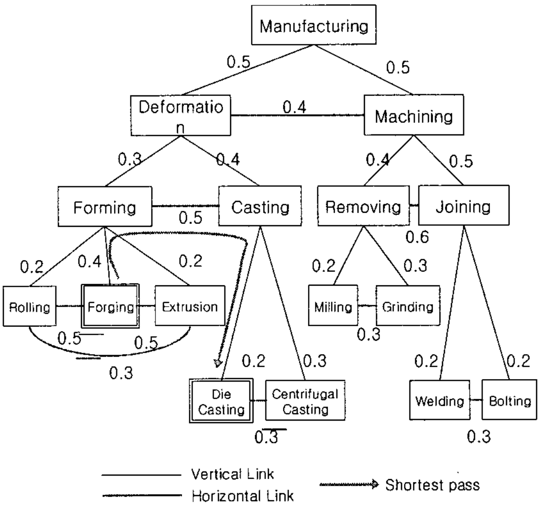


Fig. 3. Metricized knowledge abstraction hierarchy on Manufacturing.
그림 3. Manufacturing에 대한 계량화된 지식추상화 계층

To develop the distance metric that satisfies the properties above mentioned, first, we assume basic distance. The basic distance is a real number greater than 0 and not exceeding 1, and the distances are assigned by the domain expert's own decision. Of course, the basic distances are automatically derived from underlying database using the methods of data mining, neural network, and so on [9]. However, we do not focus on automatic distance evaluation in this paper. The basic distance is specified on the two types of links, vertical link and horizontal link. The vertical link connects a specific node and its 1 level abstract node, and the horizontal link connects two different 1 level neighbor nodes. Fig. 3 shows an example of the MKAH with the basic distance.

Definition 3 (*Shortest path*) **The shortest path in the MKAH is defined as the path that has the smallest number of links connecting the nodes.**

We can think the shortest path as a concatenation of the following three sub-paths: one abstraction path, one horizontal link and one specification path, where each sub-path can be null. If the horizontal link is not null, this link connects the abstraction path and the specification path, and the shortest path goes through two 1 level specific nodes of the lowest common abstract node.

Herein, we define the distance between two arbitrary nodes that satisfies the requirement of the distance metric. It is possible to consider only the basic distances on the shortest path for distance calculation however, this approach cannot always guarantee the property 2, implying that sometimes the calculated distance between the two nodes having the level difference 3 might be closer than those having the level difference 2. Therefore, the distances between two arbitrary nodes in the MKAH are formulated using the level difference and the basic distances on the shortest path.

Definition 4 (*Distance*) The distance between two arbitrary nodes x and y in the MKAH, $D(x,y)$, is defined as

$$D(x,y) = \text{level difference of } x \text{ and } y - 1 + \frac{\sum_{i=0, Z_0=x, Z_{r+1}=y}^{r} bd(Z_i, Z_{i+1})}{r+1}.$$

where, $bd(Z_i, Z_{i+1})$ is the basic distance of Z_i and Z_{i+1}, and $Z_0, Z_1, \ldots Z_{r+1}$ is the shortest path of x and y.

Lemma 1. *For two different nodes x and y in the MKAH, the distance of definition 4, D(x,y), is ranged according to the level difference n, as*

$$n - 1 < D(x,y) \le n,$$

where the n is the level difference of x and y.

Proof: In the equation in Definition 4, since the basic distance is a real number greater than 0 and not exceeding 1, the range of the third term of $\frac{\sum_{i=0, Z_0=x, Z_{r+1}=y}^{r} bd(Z_i, Z_{i+1})}{r+1}$ is greater than 0 and not exceeding 1. Hence, the distance between two nodes with level difference n, D, is ranged as

$$n - 1 < D(x,y) \le n.$$

Lemma 2. *For arbitrary nodes x, y and z in the MKAH, if the level difference of x and y is smaller than the level difference of x and z, then,*

$$D(x,y) < D(x,z).$$

Proof: Let m be the level difference of x and y, and n be the level difference of x and z, then $m-1 < D(x,y) \le m$ and $n-1 < D(x,z) \le n$. Since $m < n$,

$$D(x,y) < D(x,z).$$

The distance of Definition 4 satisfies the requirement of the distance metric. The distance between two arbitrary nodes can be determined, and the distances are grouped with respect to the level difference (Lemma 1 and 2).

2.3 Data Model and Operations

In this section we present the data model and several operations to manage the MKAH. Fig. 4 shows four

DOMAIN_ABSTRACTION

domain	abstract_domain	MKAH	level
MAJOR_GROUP	NULL	Major & Skill	4
MAJOR	MAJOR_GROUP	Major & Skill	3
EXPERTISE	MAJOR	Major & Skill	2
SKILL	EXPERTISE	Major & Skill	1

ABSTRACTION

specific_node	domain	abstract_node	distance
Deformation	MAJOR	Manufacturingl	0.5
Forming	EXPERTISE	Deformation	0.3
Casting	EXPERTISE	Deformationl	0.4
....	...	...	...

HORIZONTAL_DISTANCE

domain	node1	node2	distance
MAJOR	Deformation	Machining	0.4
EXPERTISE	Forming	Casting	0.5
SKILL	Rolling	Forging	0.5
....	...	...	...

ATTRIBUTE_DOMAIN_MAPPING

relation	attribute	domain
EMPLOYEE_SKILL	skill	SKILL
COLLEGE_MAJOR	major	Major
SKILL_FOR_TASK	required_skill	SKILLl
....	...	...

Fig. 4. Example Knowledge database for managing the Manufacturing MKAH.

그림 4. Manufacturing MAKH를 관리하기 위한 지식 데이터베이스 예제

relations (DOMAIN_ABSTRACTION, ABSTRACTION, HORIZONTAL_DISTANCE and ATTRIBUTE_DOMAIN_MAPPING), which are a knowledge database. Using these relations, we consider the following two operations.

1. *ABSTRACTNODE(n, D, l)* returns the l level abstract node of node n in domain *D*. If n is the root, which has no parent, null is returned. This function refers the relation ABSTRACTION and DOMAIN_ABSTRACTION to find the abstract node.
2. *SPECIFICNODE(n, D, l)* returns the l level specific nodes of node n in domain *D*. If n is the leaf, which has no child, null is returned. This function refers the relation ABSTRACTION and DOMAIN_ABSTRACTION to find the specific nodes.

III. Cooperative Query Answering Using MKAH

The MKAH performs query relaxation that can be invoked in various ways depending on the user's need and knowledgeof the MKAH. In the MKAH, An abstract node and its specific nodes are considered to be conceptually equal since they have an IS_A relationship. On the other hand, neighbor nodes are approximately equalsince they have the same abstract node that is conceptually equal to each neighbor nodes (Fig. 5). We present four typical query answering mechanisms: approximate selection, approximate join, conceptual selection, and conceptual join. The queries posed to an underlying database are classified according to these relaxation methods of the MKAH.

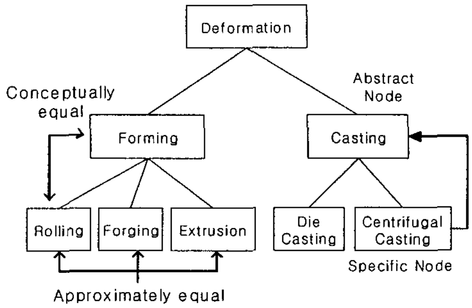


Fig. 5. Two kinds of equality for relaxing search condition.
그림 5. 검색조건 유연화의 두 종류

We use similar-to operator, of which symbol is '=?', to express cooperative queries [6, 12]. Cooperative queries are specified simply by using '=?' in the where clause of the SQL statement. For the demonstration and explanation of the query answering processes, we use a simplified personnel database that is defined as the following:

EMPLOYEE (id, name, dept, title)
EMPLOYEE_SKILL (id, skill, level)
COLLEGE_MAJOR (id, major, entrance_date, graduate_date)
SKILL_FOR_TASK (task, required_skill)
MAJOR_FOR_TASK (task, required_major)

The EMPLOYEE relation provides the current job position information of an employee, while the EMPLOYEE_SKILL relation provides the skill of employee. SKILL_FOR_TASK relation defines prerequisite skill for tasks, and MAJOR_FOR_TASK relation prescribes the relationships between individual tasks and the college major requirements for the task.

First, an approximate selection provides not only the target value itself but also its approximately equal values. For example, if. the right candidate employees who have the skill on `Milling` and `Welding` are unavailable or there are not enough, other employees with related skills need to be obtained by enlarging the scope of the search. The approximate selection query for searching for employees with the skill on `Milling` and `Welding` is written as follows:

Original Cooperative Query

```
Select  e.name, e.dept
    From EMPLOYEE e, EMPLOYEE_SKILL s
   Where s.skill =? 'Milling' and
         s.skill =? 'Welding' and
         e.id = s.id
```

Query Relaxation

Step 1.

```
where s.skill is-a
     ABSTRACTNODE('Milling','SKILL', 1)
  and s.skill is-a
     ABSTRACTNODE('Welding',' SKILL', 1)
 and e.id = s.id
```

```
⇒ where  s.skill is-a 'Removing' and
         s.skill is-a 'Joining' and
         e.id = s.id
```

Step 2.

```
  where s.skill in
       SPECIFICNODE('Removing','EXPERTISE',1)
    and s.skill in
       SPECIFICNODE('Joining','EXPERTISE',1)
    and e.id = s.id
```

```
⇒ where  s.skill in 'Grinding','Milling'}
      and s.skill in {'Welding','Bolting'}
      and e.id = s.id
```

Note that if the similar-to operator is to be meaningful, both the attribute and specified value should be in the same domain. In this sense, the approximate selection in the above query is valid since both domains are identically SKILL. The original cooperative query can be relaxed by using two operations ABSTRACTNODE() and SPECIFICNODE() defined in section 2.3. In step 1, target value Milling and Welding are generalized to Removing and Joining respectively by using the operation ABSTRACTNODE(). In step 2, by using the operation SPECIFICNODE(), the query condition is relaxed with {Milling, Grinding} and {Welding, Bolting} and the relaxed query can be answered as an ordinary SQL query. As a result of the relaxed query, the system will return the employees who have the requisite skill in Removing and Joining not only the employees who have the skill in Milling and Welding. Also they are ranked according to the semantic distance that is calculated using the equation

$$SD = \sum_{i=1}^{n} w_i \cdot D(v_t, v_r) , \quad \text{(1)}$$

where SD is a semantic distance between the target condition and the relaxed condition, and n is the number of the similar-to operator in the query, and w_i is the weight value for each condition, and t_i and r_i are the target value and relaxed value of each condition.

Second, as an extension of the approximate selection, an attribute in the join condition can be abstracted into an approximate range of nearby values. If viewed in terms of the MKAH, the query is equivalent to joining the two attributes on the basis of their abstract values, which would bring in broader join results than the join based on the ordinary specific values. The approximate join query searching for people having experienced a skill that is close to the required skill for Manufacturing Manager task is written as follow:

Original Cooperative Query

```
Select  e.name, e.dept, e.title
  From  EMPLOYEE    e,    EMPLOYEE_SKILL    s,
        SKILL_FOR_TASK t
 Where  e.id = s.id and
        t.task = "Software Design" and
        s.skill =? t.required_skill
```

Query Relaxation

Step 1.

```
  From  EMPLOYEE e, EMPLOYEE_SKILL s,
        SKILL_FOR_TASK t
 Where  e.id = s.id
   and  t.task = "Manufacturing Manager"
   and  ABSTRACTNODE(s.skill, 'SKILL',1)
     =  ABSTRACTNODE(t.required_skill,
        'SKILL' , 1)
```

Step 2.

```
 from EMPLOYEE e, EMPLOYEE_SKILL s,
      SKILL_FOR_TASK t,
      ABSTRACTION a1, ABSTRACTION a2
Where e.id = s.id
```

```
and t.task = "Manufacturing Manager"
and s.skill = a1.SPECIFIC_NODE
and t.required_skill= a2.SPECIFIC_NODE
and a1.ABSTRACT_NODE= a2.ABSTRACT_NODE
```

The use of the similar-to operator is quite analogous to the case in the approximate selection query in the sense that both compared attributes are to be in the same domain. In the above example both domains are SKILL. In step 1, query relaxation is made by abstracting the two attributes using the `ABSTRACTNODE()` operation. Strictly speaking, the `ABSTRACTNODE()` operation could not be applied to an attribute, but we just use the operation as an intermediate process to explain the query relaxation. In step 2, Since the ABSTRACTION relation provides pair of specific value and abstract value, joining the two relations on the basis of common abstract nodes can be performed using the ABSTRACTION relation as an intermediary. A relaxed ordinary query can be written as the query in step 2. Also, The distance between `s.skill` and `t.required_skill` would be calculated and the results are ranked by the semantic distances.

The conceptual selection and the conceptual join can be handled similar way to handle the approximate selection and the approximate join.

IV. Implementation

We implemented the MKAH to construct a career job search system that considers skill, salary and age of applicants. This system employed the Microsoft SQL Server 2000 as the database server, and the user interfaces were programmed with Microsoft Visual C++ 6.0. Fig. 6 shows the example of the cooperative query answering system. '`Rolling`' is selected as the target value of skill domain, and '`70`' is selected as the target value of the salary domain, and '`30`' is selected as the target value of the age domain. This will perform an approximately equal 1-level search on the skill domain and range queries for salary and age domain. The query returns the cooperative answers and the distance from target values to each answer is calculated using the equation (1). We use values 1, 10 and 3 for each skill, salary and age domain as radius value.

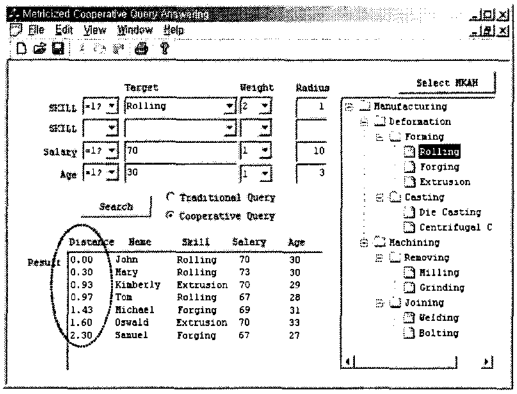


Fig. 6. Example of metricized cooperative query answering.

그림 6. 계량화된 협력적 질의 처리 예

V. Conclusions

In this paper, we have proposed the MKAH as a knowledge representation framework to support cooperative query answering which relaxes the search condition and provides approximate answers as well as exact answers. The MKAH has an abstraction-based hierarchy that supports the multi-level knowledge representations, and manages the basic distancesamong neighbor nodes. We have defined the distance metric that calculates distances between two arbitrary nodes by incorporating the basic distances. The calculated distances are grouped by the level differences, and they are precisely discriminated according to the similarity, so that we can get consistency in the distance metric among all nodes in the MKAH. Since the MKAH considers only neighbor values when managing the distances, while the tabular methods consider all the other values, maintenance cost is considerably reduced with the MKAH.

In terms of cooperative query answering, the MKAHcan guide more interactive and flexible query relaxation with the abstraction level information. Since the MKAH supports the distance metric, it is appropriate to handle the quantitative similarity of categorical data, and corresponding query results are ranked according to the distances from query conditions. Also, the domain adoptingthe MKAH could be compatible with other numeric domains when calculating the distance between target record and neighbor records.

Finally we have presented the data model and operations to implement the MKAH, and have implemented a prototype cooperative query answering system to evaluate the effectiveness of the proposed framework. Our preliminary experimental results reveal that the MKAH provides rich semantic representation and a high quality distance metric and is appropriate for building up a scalable and extendable system by integrating the semantic distance approach.

The MKAH can be appropriate for a knowledge representation framework of the systems manipulating nominal data as well as numeric data, such as, knowledge management systems, search engines, decision support systems and so on.

For future research, systematic procedures to construct the MKAH should be conceived for practical application. In addition, we will study the context sensitive MKAH which modulate the basic distance values as context changes.

Reference

[1] Abiteboul S., Benjelloun O. & Milo T. (2002) Web services and data integration. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering*, pp 3-6, Singapore.

[2] Babcock B., Chaudhuri S. & Das G. (2003) Dynamic Sample Selection for Approximate Query Processing. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pp 539-550, San Diego, California, USA.

[3] Barg M. & Wong R.K. (2000) A Multi-Agent Architecture for Cooperative Query Answering. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, pp 3138-3147, Maui, Hawaii, USA.

[4] Bernardino J., Furtado P. & Madeira H. (2002) Approximate Query Answering Using Data Warehouse Striping. *Journal of Intelligent Information Systems* 19(2), 145-167.

[5] Braga J.L., Laender A.H.F. & Ramos C.V. (2000) A Knowledge-Based Approach to Cooperative Relational Database Querying. *International Journal of Pattern Recognition and Artificial Intelligence* 14, 73-90.

[6] Chu W., Yang H., Chiang K., Minock M., Chow G & Larson C (1996) CoBase: A scalable and extensible cooperative information system. *Journal of Intelligent Information Systems* 6(2/3), 223-259.

[7] Cuzzocrea A. & Matrangolo U. (2004) Analytical Synopses for Approximate Query Answering in OLAP Environments. In *Proceedings of the 15rd International Conference on Database and Expert Systems Applications*, pp. 359-370, Zaragoza, Spain,.

[8] Gaasterland T. (1997) Cooperative Answering through Controlled Query Relaxation. *IEEE Expert* 12(5), 48-59.

[9] Hong T. & Han I. (2002) Knowledge-based data mining of news information on the Internet using cognitive maps and neural networks. *Expert Systems with Applications* 23(1), 1-8.

[10] Huh S.Y. & Lee J.W. (2001) Providing Approximate Answers Using a Knowledge Abstraction Database. *Journal of Database Management* 2(2), 14-24.

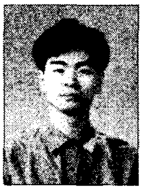
[11] Liu Q. & Ng P.A. (1998) A Query Generalizer for Providing Cooperative Responses in an Office Document System. *Data and Knowledge Engineering* 27(2), 177-205.

[12] Motro A., Anokhin P. & Berlin J. (2002) Intelligent Methods in Virtual Databases. In *Proceedings of the Fourth International Conference on Flexible Query Answering Systems*, pp 580-591, Warsaw, Poland.

[13] Tanasescu A. (2005) Cooperative Query Answering for RDF. In *Proceedings of the 15th International Syposium on Methodologies for Intelligent Systems*, pp. 535-543, New York, USA.

[14] Wang C., Li J. & Shi S. (2004) Cell Abstract Indices for Content-Based Approximate Query Processing in Structured Peer-to-Peer Data Systems. In *Proceedings of the 6th Asia-Pacific Web Conference*, pp. 269-278, Hangzhou, China.

저 자 소 개



신 명 근

2006년 2월 : KAIST 정보및통신 공학 박사

2002년 ~ 현재 : (주)투이컨설팅

관심분야 : 협력적 질의 처리, 데이터베이스, 웹구조화