APPROXIMATE QUERY ANSWERING APPROACH BASED ON DATA ABSTRACTION AND FUZZY RELATION

Soon-Young Huh[1] and Kae-Hyun Moon[1]

Graduate School of Management[1]
Korea Advanced Institute of Science and Technology
207-43 Chongyangni-dong, Dongdaemoon-gu, Seoul, Korea 130-012

ABSTRACT

Although a query language has been used as a convenient tool to obtain information from a database, users increasingly demand more intelligent cooperative query answering system that can understand the intent of an imprecise query and provide additional useful information as well as exact answers. This paper proposes an integrated cooperative query answering approach that is based on the notion of data abstraction and fuzzy relation. The approach help database users carry out more effective query answering and decision making by increasing the diversity of admitted queries, elaborating the query relaxation control through user interaction, and accommodating dynamic knowledge maintenance.

KEYWORDS

Approximate Query, Query Relaxation, Financial Decision Support Systems

1. Introduction

Although query language has been widely used as a convenient tool to obtain information from a database, users increasingly demand more user-friendly and fault-tolerant query answering systems that can understand the intent of an imprecise query and provide additional useful information as well as exact answers. This is largely due to exactness in nature of the relational databases and the query languages.

Traditional query language, as a standard database user interface, has often frustrated database users including expert users by enforcing rigidity and preciseness in query writing with respect to query syntax and database schema. Even though the query syntax is correct, it provides only exact answers. When exact answers are unavailable or insufficient, users must rewrite an appropriate query for themselves to produce approximate answers. Thus, to get satisfactory result, database users are required to precisely understand the database schema, underlying data, and problem domain, which is not easy for ordinary users.

To remedy such shortcomings and to enhance the effectiveness of information retrieval, users need fault-tolerant and intelligent database interface that allows users to express vague queries, analyzes the intent of a vague query and provide additional useful information as well as exact answers. For this purpose, a variety of CQA approaches have been proposed. However, existing approaches have limitations in the diversity of vague queries and elaborate query relaxation control through user interaction because they do not have rich data semantic knowledge. Also, most studies have focused on the construction of the knowledge representation framework and the query answering process but insufficiently addressed the issue of the semantic knowledge maintenance.

In this paper, we first develop knowledge representation framework that has rich semantics by integrating data abstraction and fuzzy relation approaches. The framework represents semantic relationships between data values and meta-data necessary for query relaxation. And then, we classify all possible cooperative queries and develop query processing mechanism supporting all kinds of queries in a consistent and integrated manner. The CQA can be invoked in various ways depending on the query requirements and the extent of the query relaxation. We present four typical query answering mechanisms using the knowledge abstraction database.

The organization of this paper is as the following: In chapter 2, we introduce a knowledge representation framework, and in chapter 3, we construct knowledge abstraction database for query answering. In chapter 4, we develop CQA mechanism. Finally, chapter 5 summarizes our research contributions.

2. Fuzzy Knowledge Abstraction Hierarchy

We introduce fuzzy knowledge abstraction hierarchy (FKAH), a knowledge representation framework that integrates data abstraction and metric approaches for representing semantic relationship between data values.

2.1 Notion of Data Abstraction in the Framework

The FKAH uses *data abstraction* that has been considered as an effective method to accommodate the semantic relationships between data values [25]. In cooperative query processing, such data abstraction is also useful to associate data values in the process of query relaxation.

To illustrate the FKAH, Figure 1 shows two FKAH instances derived from a same underlying database. While the upper one is College Major hierarchy classifying data on the majors of employees into multiple abstraction levels including major name, major area, and major group, the lower one is Career Development Education hierarchy on the education course. Values constituting the hierarchies may be part of the underlying database or artificial values added to describe the semantic relationship among the existing data values.

The FKAH is composed of two types of abstraction hierarchies: *value abstraction hierarchy* and *domain abstraction hierarchy*. First, in the value abstraction hierarchy, a *specific value* is generalized into an *abstract value* and the abstract value can be generalized further into a more abstract value. Conversely, a specific value is considered as a specialized value of the abstract value. Thus, a value abstraction hierarchy is constructed on the basis of generalization/specialization relationships between abstract values and specific values in various *abstraction levels*, which is obtained by using value abstraction. The value abstraction relationship can be interpreted as IS-A relationship. For instance, Finance is a (major name of) Management while Management is a (major area of) Business. As such, higher levels provide a more generalized data representation than lower ones.

While the cardinal relationship between an abstract value and its specific values is assumed to be one-to-many, a specific value can also have multiple abstract values that are located in different abstraction levels along a path from the specific value to its most abstract value at the highest abstraction level. In such capacity, an abstract value is called *n-level abstract value* of the specific value according to the abstraction level difference *n*.

Second, the domain abstraction hierarchy consists of *domains* that encompass all individual values in a value abstraction hierarchy and there exist INSTANCE-OF relationships between the domains and values. Much as generalization/specialization relationships exist between the data values in two different abstraction levels of the value abstraction hierarchy, a *super-domain/sub-domain* relationship exists between two different domains, which is

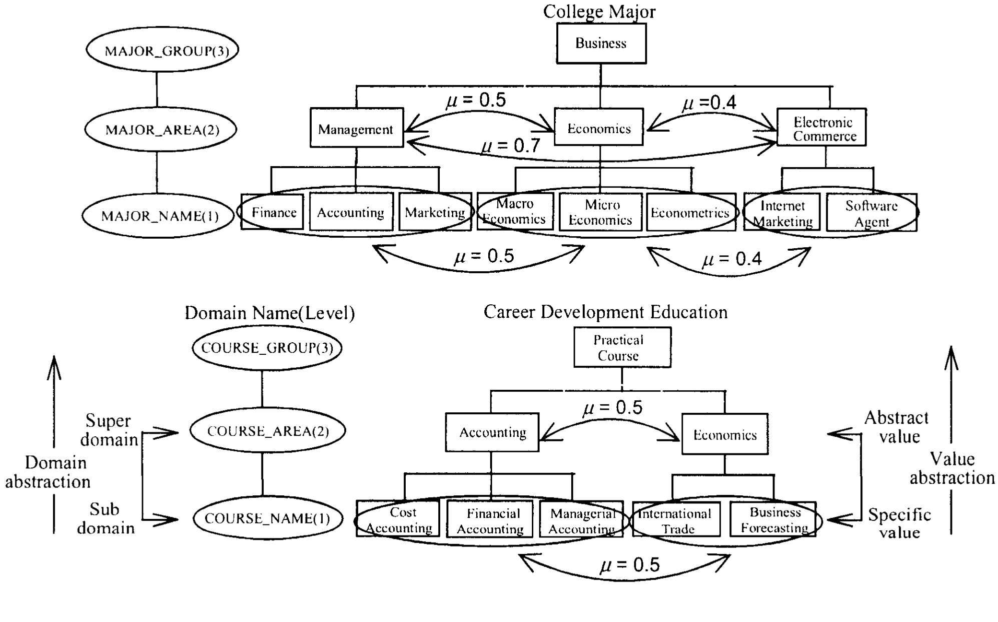

Figure 1. Example of FKAH Instances

obtained by *domain abstraction*. For instance, MAJOR_AREA is the super-domain of MAJOR_NAME. All the abstract values of instance values in a sub-domain correspond to the instance values of the super-domain. Thus, the super-domain MAJOR_AREA is more generalized than sub-domain MAJOR_NAME, since MAJOR_AREA contains more generalized values than MAJOR_NAME. The cardinal relationship between two adjacent domains is assumed to be one-to-one and a super-domain is called *n*-level super-domain of the sub-domain according to the abstraction level difference *n*.

Multiple FKAH instances, as shown in Figure 1, can exist for a single underlying database when multiple perspectives are required for the database. Though a value can be located in multiple hierarchies and thus, in multiple domains, a domain should be unique and be located only in one hierarchy. The Economics value in Figure 1, for instance, is found in both College Major and Career Development Education hierarchies, and belongs to different domains depending on the hierarchy. It belongs to the MAJOR_AREA domain in the College Major hierarchy, while the COURSE_AREA domain in the Career Development Education hierarchy. In this sense, as a value can exist in multiple hierarchies, its abstract values and specific values are not uniquely determined by the value itself. Domain information is additionally needed to identify the abstract and specific values for a given specific value. We call this property the *domain-dependency* of abstract values and specific values.

2.2 Notion of Fuzzy Relation in the Framework

We use the fuzzy relation to add the metric approach to the knowledge abstraction hierarchy based on the data abstraction approach. A relation is any subset of a Cartesian product of two sets. For instance, a subset of A×B, called a (binary) "relation from A to B", is a collection of ordered pairs (a, b) with first components form A and second components from B, and, in particular, a subset of A×A is called a "relation on A". For a binary relation R, one often writes *a R b* to mean that (a, b) is in R. Thus, when there exists a similarity relationship between values v_1 and v_2, we can represent them as an ordered pair (v_1, v_2) and a relation comprised of the pairs. There exists a relation R in College Major Hierarchy as follows.

R={(Finance, Accounting), (Finance, Marketing), (Accounting, Marketing), (Micro Economics, Macro Economics), (Micro Economics, Econometrics), (Macro Economics, Econometrics), (Internet Marketing, Software Agent), (Management, Economics), (Management, Electronic Commerce), (Economics, Electronic Commerce)}.

Fuzzy relation R is suggested by defining a membership degree function μ_R: A×B → [0,1] for each pairs that can be interpreted as the membership degree or strength of the relation. In other words, when $\mu_R(v_1, v_2)$ is 0.7 and $\mu_R(v_1, v_3)$ is 0.5, the similarity relation between v_1 and v_2 is stronger than that between v_1 and v_2.

Using the fuzzy relation, we can define the notion of similarity between values in the FKAH. The notion enables us to represent that d_1 and d_2 are more similar than d_2 and d_3 though they have a same abstract value e_1. In other words, more elaborate representation of similarity is possible. The assumptions can be relaxed to the following assumptions.

- $\mu_R(v_1, v_2)$ can be different from $\mu_R(v_1, v_3)$ though v_1, v_2, v_3 has same abstract value.

- $\mu_R(v_1, v_2) > \mu_R(v_1, v_3)$ iff v_1, v_2 has same 1-level abstract value and v_1, v_3 has different 1-level abstract value
- $\mu_R(v_1, v_2)$ also implies similarity degree between the sets of specific values of v_1 and v_2.

The first assumption means that the degrees of similarity between the values having a same abstract value can be different from one other. The second assumption means that the similarity between the values having a same abstract value is greater than that between the values having different abstract values. While $\mu_R(v_1, v_3)$ means the degree of similarity between v_1 and v_3, it also means the degree between the set of their specific values. In other words, $\mu_R(d_1, d_2) = 0.7 \rightarrow \mu_R(\{a_1, a_2, a_3\}, \{b_1, b_2, b_3\}) = 0.7$.

We can satisfy the assumptions by defining the degrees only for pairs of values with same abstract value. Also, by assumption 3, the similarity degree between the values having different abstract values can be derived from their abstract values. Thus, we can represent the fact that a_1 and b_1 can be more similar than a_1 and c_1.

3. Relational Schema Design of the Abstraction Database

To use the semantics of the FKAH for intelligent query processing, it was incorporated into an abstraction database. A sample abstraction database accommodating the FKAH instances in Figure 1 is provided in Figure 2, which comprises three relations: DOMAIN_ABSTRACTION, VALUE_ABSTRACTION, ATTRIBUTE_MAPPING.

Due to the domain dependency of abstract values, to get an abstract value, we must know both the value and its domain. Thus, in the VALUE_ABSTRACTION relation, both names of Value and Domain become the composite key to determine the corresponding 1-level abstract value in the Abstract_Value attribute. For instance, the abstract value of Accounting in the MAJOR_NAME domain (of the College Major hierarchy) is not Practical Course in COURSE_GROUP, but Management in MAJOR_AREA. Similarly, the siblings of Accounting are Finance and Marketing which are all in the MAJOR_NAME domain and have the same abstract value, Management.

The DOMAIN_ABSTRACTION relation is specifically indispensable when it comes to obtaining a n-level abstract value. For instance, in the VALUE_ABSTRACTION relation in Figure 2, the 2-level abstract value for Econometrics cannot be obtained by performing union operation on only 1-level abstraction pair twice since two Economics (one with MAJOR_AREA and the other with COURSE_AREA) are obtained in the VALUE_ABSTRACTION relation.

To identify the right value, we additionally need the super-domain of the current domain of Econometrics (i.e., MAJOR_NAME). Thus, from the DOMAIN_ABSTRACTION relation, we can identify MAJOR_AREA to be the super-domain inquired and thus use Economics and MAJOR_AREA as the composite key value in the VALUE_ABSTRACTION relation to determine Business as 2-level abstract value for Econometrics.

DOMAIN_ABSTRACTION

Domain	*Super_Domain*	*Hierarchy*	*Abstraction_Level*
MAJOR_NAME	MAJOR_AREA	College Major	1
MAJOR_AREA	MAJOR_GROUP	College Major	2
MAJOR_GROUP		College Major	3
COURSE_NAME	COURSE_AREA	Career Development Education	1
COURSE_AREA	COURSE_GROUP	Career Development Education	2
COURSE_GROUP		Career Development Education	3

VALUE_ABSTRACTION

Value	*Domain*	*Abstract_Value*
Finance	MAJOR_NAME	Management
Accounting	MAJOR_NAME	Management
Marketing	MAJOR_NAME	Management
Macro Economics	MAJOR_NAME	Economics
Micro Economics	MAJOR_NAME	Economics
Econometrics	MAJOR_NAME	Economics
Management	MAJOR_AREA	Business
Economics	MAJOR_AREA	Business
Business	MAJOR_GROUP	
Cost Accounting	COURSE_NAME	Accounting
Financial Accounting	COURSE_NAME	Accounting
Managerial Accounting	COURSE_NAME	Accounting
International Trade	COURSE_NAME	Economics
Business Forecasting	COURSE_NAME	Economics
Accounting	COURSE_AREA	Practice Course
Economics	COURSE_AREA	Practice Course
Practice Course	COURSE_GROUP	

ATTRIBUTE_MAPPING

Relation	*Attribute*	*Domain*
EMPLOYEE_MAJOR	MAJOR	MAJOR_NAME
CAREER_PATH	Task	COURSE_NAME
CAREER_PATH	Prerequisite_Task	COURSE_NAME
TASK_MAJOR	Task	COURSE_NAME
TASK_MAJOR	Required_Major_Area	MAJOR_AREA
TASK_HISTORY	Task_Performed	COURSE_NAME
....	....	....

FUZZY_RELATION

Value1	Value2	Degree
Finance	Accounting	0.7
Accounting	Marketing	0.4
Finance	Marketing	0.5
Macro Economics	Micro Economics	
Micro Economics	Econometrics	
Macro Economics	Econometrics	
Management	Economics	
Cost Accounting	Financial Accounting	
Financial Accounting	Managerial Accounting	
Managerial Accounting	Cost Accounting	
International Trade	Business Forecasting	
Accounting	Economics	

Figure 2. The Constructor Relations of the Abstraction Database.

4. Approximate Query Answering

. In a personnel database example, suppose that a personnel manager wants to find out people majoring in Finance to fill a certain vacant task position. If the right candidate employees, having majored in Finance, are unavailable or insufficient for a certain personnel management task, other employees with related majors may be obtained by enlarging the scope of the search. In the FKAH, searching approximate values for a specialized value is equivalent to finding the abstract value of the specialized value, since specialized values of the same abstract value constitute approximate values of one another. In the query, the relaxation operator, =?(1:6), is used to specify the relaxation requirement of the search condition. The selection condition, c.major=?(1:6) “Finance”, is to find the employee major records whose major attribute value is either “Finance” or an approximate neighborhood value within the specified approximation degree.

Approximate Selection

Step 1. Original Query

```
select   e.emp_name, e.dept
from     employee e, college_major c
where    c.major =(1:6) "Finance" and e.id = c.id
```

Step 2. Query Generalization

```
select   e.emp_name, e.dept
from     employee e, college_major c
where    c.major is-a Generalize("Finance", 1:6) and e.id = c.id

select   e.emp_name, e.dept
from     employee e, college_major c
where    c.major is-a "Management" or c.major is-a "Electronic Commerce"
         and e.id = c.id
```

Step 3. Query Specialization

```
select   e.emp_name, e.dept
from     employee e, college_major c
where    c.major in Specialize("Management") or
         c.major in Specialize("Electronic Commerce") and e.id = c.id

select   e. emp_name, e.dept
from     employee e, college_major c
where    c.major   in   ("Finance",   "Accounting",   "Marketing",   "Internet
         Marketing", "Software Agent") and e.id = c.id
```

Figure 2. Steps of Approximate Selection Query

5. Conclusions

In this paper, we propose a CQA approach that relaxes the search condition and provides approximate neighborhood information when exact answers are unavailable. For this purpose, a knowledge representation framework, namely FKAH, is developed. The FKAH is a multilevel knowledge representation framework that extract abstract values and domains from an underlying corporate database into several hierarchies using value and domain abstraction. Additionally, it adopts fuzzy relation theory to enrich the semantic data relationship. Thus, in FKAH, a similarity relationship between data values is represented based on the integrated notion of data abstraction and fuzzy relation. We also presented a query processing mechanism handling a variety of cooperative queries including approximate queries and conceptual queries in a consistent and integrated way. The proposed FKAH increases the diversity of admitted cooperative queries and supports more interactive and flexible query relaxation processes.

REFERENCES

1. Chen,Q., Chu, W., and Lee, R. (1990) Providing Cooperative Answers via Knowledge-Based Type Abstraction and Refinement. Proc. of the 5th International Symposium on Methodologies for Intelligent Systems.
2. Chu, W., Yang, H., Chiang, K., Minock, M., Chow, G. and Larson, C. (1996) CoBase: A Scalable and Extensible Cooperative Information System. International Journal of Intelligence Information Systems. 6
3. Chu, W., Yang, H., and Chow, G. (1996) A Cooperative Database System (CoBase) for Query Relaxation. Proc. of the Third International Conference on Artificial Intelligence Planning Systems
4. Chu, W. and Chen, Q. (1994) A Structured Approach for Cooperative Query Answering. IEEE Transactions on Knowledge and Data Engineering, 6(5), 738-749
5. Chu, W., Lee, R., and Chen, Q. (1991) Using Type Inference and Induced Rules to Provide Intensional Answers. Proc. of the 7th International Conference on Data Engineering, 396-403
6. Chu, W., Chen, Q., and Lee, R. (1991) Cooperative Query Answering via Type Abstraction Hierarchy. Cooperative Knowledge Base System. North-Holland, Elservier Science Publishing Co., Inc.
7. Cuppens, F. and Demolombe, R. (1989) Cooperative Answering: A Methodologies to Provide Intelligent Access to Databases. Proc. of 2nd International Conference on Expert Database Systems, 621-643
8. Godfrey, P., Minker, J., and Novik, L. (1994) An Architecture for a Cooperative Database System. Proc. of the 1994 International Conference on Applications of Databases
9. Hemerly, A., Casanova, M., and Furtado, A. (1994) Exploiting User Models to Avoid Misconstruals. Nonstandard Queries and Nonstandard Answers, Oxford Science Publications
10. Minock, M. J. and Chu, W. (1996) Explanation for Cooperative Information Systems. Proc. of Ninth International Symposium on Methodologies for Intelligent Systems
11. Motro, A. (1990) FLEX: A Tolerent and Cooperative User Interface to Databases. IEEE Transactions on Knowledge and Data Engineering, 2(2), 231-246
12. Motro, A. (1994) Intensional Answers to Database Queries. IEEE Transactions on Knowledge and Data Engineering, 6(3), 444-454
13. Motro, A. (1990) Accommodating Inprecision in Database Systems: Issues and Solutions. Data Engineering, 13(4) 29-34
14. Shenoi, S. and Melton, A. (1992) Functional Dependencies and Normal Forms in the Fuzzy Relational Data Model. Information Sciences, 1-28
15. Takahashi, Y. (1993) Fuzzy Database Query Languages and Their Relational Completeness Theorem. IEEE Trans. Knowledge and Data Engineering, 5(1), 122-125
16. Umano, M. and Ezawa, Y. (1991) Implementation of SQL-type Data Manipulation Language for Fuzzy Relational Databases. Proc. of IFSA Brussels.
17. Vrbsky, S. V. and Liu, W. S. (1993) APPROXIMATE-A Query Processor that Produces Monotonically Improving

Approximate Answers. IEEE Transactions on Knowledge and Data Engineering, 5(6)

18. Wong, M. H. and Leung, K. S. (1990) A Fuzzy Database Query Language. Information Systems, 15(5), 583-590