Providing Approximate Answers Using Data Abstraction, Fuzzy Relation, and Thesaurus

Kae-Hyun Moon and Soon-Young Huh

Graduate School of Management, Korea Advanced Institute of Science and Technology
207-43 Chongyangni-dong, Dongdaemoon-gu, Seoul, Korea 130-012
Tel: +82-2-958-3626, Fax: +82-2-958-3604, E-mail: khmoon@kgsm.kaist.ac.kr

Abstract

The objective of this paper is to propose a cooperative query answering approach that relaxes query conditions and provides approximate answers by utilizing semantic relationships between data values. The proposed fuzzy abstraction hierarchy (FAH) adopts the integrated notion of data abstraction and fuzzy relation to represent similarity relationship. Also, to be tolerant of incorrect input for queries, FAH refers to a thesaurus for synonym information about database token. For query processing based on FAH, query relaxation operators such as query generalization, approximation, and specialization are developed. Compared with existing approaches, FAH supports more effective information retrieval by processing various kinds of cooperative queries through elaborate relaxation control, and by providing ranked query results according to fitness scores.

Keywords:

Approximate answer; Data abstraction

1. Introduction

Traditional database query processing often cannot provide the information the users really expect because the conventional database systems do not have any intelligence to produce answers cooperatively with users. Database systems return null responses when the exact answers to queries do not exist. Non-empty responses also may not meet users' expectation that requires not only exact answers but also extra approximate answers. If a query processing system understands the schema and semantics of the database, it will be able to produce such informative responses and greatly help the user write intended queries. To support such intelligent query processing, a number of cooperative query answering approaches have been introduced into conventional database systems, which provide a human-oriented interface to a database system by facilitating query relaxation for producing approximate answers [3-11, 13-14, 17-19].

The semantic distance approach [17-19] represents the degree of similarity between a pair of data objects by a numerical distance. This approach has advantages of ease and efficiency in developing query relaxation algorithms, since quantitative distances among data objects are easy to compute. The abstraction hierarchy approach [3-7] applies the data abstraction method to object instance and replaces the instance attribute values by the abstract concepts to which they belong. The abstract representation forms a clustering hierarchy where similar attribute values are clustered by the abstract concepts and the individual abstract concepts are related to one another by a certain abstraction function or the abstraction similarities. The abstraction hierarchy approach is specifically advantageous when the characteristics of data objects are qualitative and categorical. However, each approach is yet to be satisfactory in terms of the diversity and proximity measurability for effective cooperative query answering. For query diversity, the knowledge representation approach should support users of different levels of expertise in information retrieval. According to their expertise, users express their requests with diverse kinds of cooperative queries. Thus, the ability to processing diverse cooperative queries is necessary for handling various requests of the diverse users.

For query proximity measurability, the approach should provide a measure to help users determine the priority of each approximate answer (e.g., fitness score) in the query result set. For this, we need a measure of similarity relationship between data values. With such a measure, users can obtain ranked result sets that they can use to perform elaborate and flexible query relaxation control through user interaction.

To overcome these limitations, this paper proposes a new knowledge representation framework called fuzzy abstraction hierarchy (FAH) that integrates the abstraction hierarchy and the semantic distance approach. Also, to be tolerant of incorrect input for queries, FAH adopts thesaurus for synonym information about database token. Based on FAH, we develop query relaxation operators including generalization, specialization, and approximation of data values. This paper is organized as follows: Section 2 reviews prior related approaches. Section 3 introduces FAH. In Section 4, we develop basic query relaxation operations for cooperative query answering and introduce a thesaurus to be tolerant of incorrect input for queries. Finally, Section 5 concludes the paper.

2. Overview of Related Approaches

Query relaxation is performed by associating the values in the query condition to other related values on the basis of predefined semantic relationships between data values. Thus, an appropriate knowledge representation framework is needed for human experts to extract and manage the semantic relationships between data values. Among previous approaches, the semantic distance and the abstraction hierarchy approaches are commonly used to represent the semantic knowledge from the underlying database.

2.1 Semantic Distance Approach

The semantic distance approach introduced the notion of distance, i.e., scalar values, to measure the strength of similarity between scalar data values. Generally, there exist two kinds of data values in databases: quantitative and qualitative. For quantitative data values, the distances can be derived by computation based on numerical operators, e.g., the absolute value of the difference [17, 18]. For qualitative data values, the approach determines the distance based on *semantic distance* between two values. Even though qualitative data values may not have an explicit numerical semantic relationship, the expert administrator may establish the semantic distance between the values in numerical terms that can be stored in a table. In this case, the distances are assigned by the domain expert [17].

Since numerical distances between data values are easy to compute, this approach has the advantage of ease and efficiency in developing query relaxation algorithms. Also, the distances can be used as a measure to determine the rank of each answer, which helps users find useful information related to the retrieved answers. However, this approach is limited due to the difficulty in transforming quantitative and qualitative data values into a uniform numerical measure. In addition, it is difficult to objectively and consistently assess the similarity distance between qualitative data values because there are no supplementary criteria besides the expert's subjective criteria. In other words, the similarity distances are totally determined by the expert's subjective criteria only. Since semantic distances need to be established for every value pair, the number of value pairs to manage quickly becomes unmanageable with real world application systems. As a result, maintenance cost increases and the consistency of the distance measure is jeopardized.

2.2 The Abstraction Hierarchy Approach

The abstraction hierarchy approach adopts the abstract representation of data values, and replaces the values by the abstract concepts to which they belong. The abstract representations form a data abstraction hierarchy where similar values are clustered by the abstract concepts and the individual abstract concepts are related to one another by a certain abstraction function or by abstraction similarities. Thus, this approach is specifically advantageous when the data values are qualitative and categorical.

To relax a query condition, this approach exploits query *abstraction/refinement* methods that rely on the value abstraction hierarchy. A query abstraction is accomplished by replacing data values from the query with more abstract values from the hierarchy, and the resulting query is considered more general with relaxed condition than the original. The general query is then refined into a set of specific queries to be evaluated against the underlying database.

The knowledge abstraction hierarchy (KAH) [13, 14] extended previous abstraction hierarchy approaches by capturing not only value abstraction but also domain abstraction knowledge from underlying databases. KAH can support more effective query processing by increasing the diversity of admitted queries and by accommodating dynamic abstraction knowledge maintenance.

In KAH, three semantic relationships exist between data values (i.e., parent, child, and siblings viewed in terms of a tree structure): abstraction, specification, and approximation.

- Generalization/Specialization

A *specific value* (i.e., child) in a lower *abstraction level* is abstracted (generalized) into an *abstract value* (i.e., parent) in a higher abstraction level and the abstract value can be abstracted further into a more abstract value. The abstract value is called *n-level abstract value* of the specific value where n is the difference in the abstraction levels of the values. The highest abstraction level is the *depth* of the hierarchy.

- Approximation

Among the values existing in the same abstraction level, *n-level siblings* are defined to be composed of values whose nearest same abstract value is n-level abstract value.

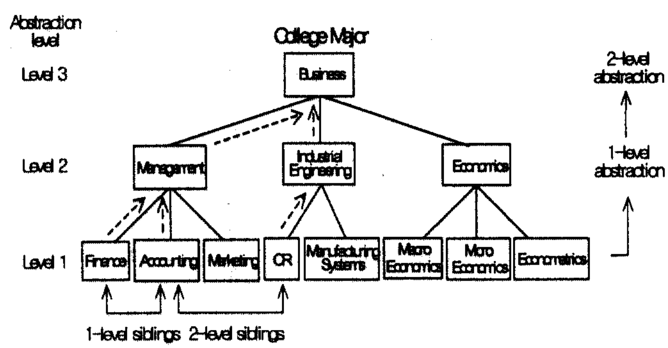

Figure 1 - Example of College Majors as a KAH.

Figure 1 shows an example of KAH of College Major. Management is 1-level abstract value of Finance, and Business is 2-level abstract value of Finance. Conversely, Finance is 1-level specific value of Management and 2-level specific value of Business. Finance, Accounting, and Marketing are 1-level siblings because their nearest same abstract value is 1-level abstract value Management. Accounting and OR are 2-level siblings. In KAH, the approximate values of a value and their similarity distance are described as follows:

- *Monotonously Increasing Range of Approximate Values.*

The *n-level approximate values* of a value are restricted to the values ranging from its 1-level siblings to n-level siblings that have the same n-level abstract value where $n \geq$

1.

- *Uniform Similarity Strength.*

The similarity distances between the approximate values are uniform.

The range of approximate values depends on the abstraction level and can be controlled by adjusting the abstraction level. For example in Figure 1, the approximation of Finance returns {Accounting, Marketing} within 1-level abstraction because they have the same 1-level abstract value Management, but it can be extended to {Accounting, Marketing, OR, Manufacturing Systems, Macro Economics, Micro Economics, Econometrics} within 2-level abstraction. In other words, Finance and OR are not approximate values of Finance within 1-level abstraction, but they become approximate values of Finance within 2-level abstraction as a result of extension of the range by increasing the abstraction level.
Like other data abstraction approaches, KAH does not define the measure of the semantic relationship between the values in a hierarchy. Thus, the approximate values have uniform similarity distances with one another because the data abstraction approaches cannot discriminate the similarity distance between the values (*the uniform similarity strength property*). Although the range of approximate values can expand as the abstraction level increases, the similarity distances between the values in the range are always uniform. For example, when approximate values of Finance are requested within 1-level abstraction, every 1-level sibling should be returned including Marketing and Accounting. In the same way, {Finance, Accounting, OR, Manufacturing Systems, Marketing, Macro Economics, Micro Economics, Econometrics} are uniformly approximate with one another within 2-level abstraction.
Because of the uniform similarity strength property, the data abstraction approach cannot provide query results ranked by the similarity distances (i.e., fitness scores) from the target value in the query condition. Users themselves should determine the importance of each answer in the query result without any assistance. Also, the data abstraction approach has limitations in flexible query relaxation control that require the discrimination of similarity distance between approximate values. Specifically, the uniformness property may produce too many query results in each relaxation step because the approximation of a value cannot selectively return its approximate values according to the similarity strength.

2.3 Other Approaches

The rule-based approach [8, 12] adopts first-order logic as its formal framework and delineates semantic information about data objects and data integrity constraints using first-order formulas over a set of (base, built-in, derived) predicates. In this approach, the entire database is understood as a set of base predicates, and a database query also consists of a predicate rule whereby searching information is specified with free variables. The query is answered through conflict resolution and inference mechanisms, and query relaxation is carried out by coordinating the integrity constraints. The weaknesses of this approach include a lack of systematic organization for guiding the query relaxation process and the less intuitive query answering process.
The fuzzy database approach [23] supports various kinds of fuzziness derived from the data itself and linguistic queries. It assumes that data objects in each domain can be assigned a degree of similarity between 0 and 1. In addition, the knowledge base can store various kinds of imprecise information such as mixed hair color (i.e., 0.6 brown and 0.4 black) and a certain person's residence (i.e., Boston or New York). Users compose approximate queries by using fuzzy comparators such as much-greater-than. Relations are extended to accept fuzzy values or to allow values that are sets of domain objects.

3. The Fuzzy Abstraction Hierarchy

To remedy the limitations of existing approaches, we suggest a knowledge representation framework FAH that integrates the abstraction hierarchy and semantic distance approach. Like KAH, FAH contains three semantic relationships between data values (i.e., parent, child, and siblings viewed in terms of a tree structure) – abstraction, specification, and approximation.

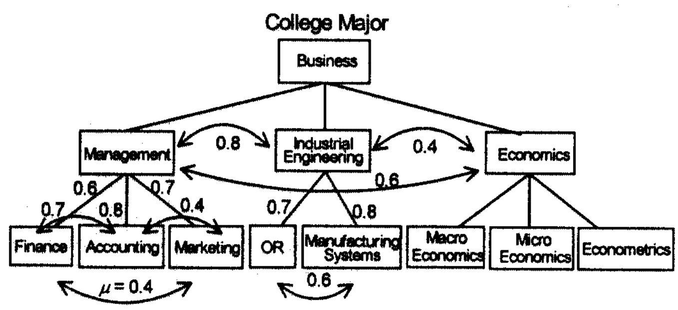

Figure 2 - Example of College Majors as a FAH.

FAH permits the discriminated similarity strength between values by adopting the 'fuzzy relation' instead of 'similarity distance' of the semantic distance approach. Originally, fuzzy relations have been used for determining the strength of the relationship between two objects in fuzzy sets. It provides the theoretical basis and fuzzy operators for deriving similarity strengths and reduces the cost of constructing and maintaining a consistent structure of similarity strengths. Thus, the fuzzy relation in FAH is defined as follows: The membership function on R, μ_R: V × V → [0, 1], where V is the set of values in FAH and R is similarity relation between the values. The number μ_R will be referred to as the *similarity strength* between data values. For example, 'μ_R(Finance, Accounting) = 0.7' in Figure 1 means that similarity strength between 'Finance' and 'Accounting' is 0.7, and because μ_R(Finance, Accounting) is greater than μ_R(Finance, Marketing), 'Finance' is more similar with 'Accounting' than with 'Marketing'. The following two propositions are established to adapt the discriminated similarity strength between values in FAH.

Proposition 1. *Monotonously Decreasing Strength*
The similarity strength between (n+1)-level siblings are smaller than those between n-level siblings, where $n \geq 1$.

Proposition 2. *Influence of Parents upon Similarity Strength*
The similarity strength between 1-level siblings influences the similarity strength between their child values.

According to proposition 1, the similarity strength between n-level siblings monotonously decreases as the abstraction level n increases. For example, since 'Finance' and 'Accounting' are 1-level siblings, and 'Finance' and 'OR' are 2-level siblings, the similarity strength between 'Finance' and 'Accounting' must be greater than that between 'Finance' and 'OR'. Proposition 1 can be utilized as a supplementary criterion for measuring the similarity strength, while semantic distance approaches do not have any supplement criteria.
A great number of value pairs make it difficult to satisfy proposition 1, which happens when every pair of values in the hierarchy should be considered for determining their strengths. The fuzzy relation approach that FAH adopts in lieu of the semantic distance approach provides the theoretical basis for decreasing the number of pairs to explicitly consider, while at the same time the similarity strength for every pair of values can be obtained. From the structural features of the FAH constructed on the basis of the data abstraction, we will develop a method and fuzzy operator that performs the composition of the fuzzy relations between data values.
On the basis of proposition 2, the expert administrator does not need to explicitly measure the strength between the n-level siblings, where $n \geq 2$. He just needs to measure the *primitive strength* - the strength between 1-level siblings (i.e., *sibling strength*) and the strength between parent and child (i.e., *abstraction strength*) as shown in Figure 2. Thus, 'similarity relation' in FAH can be defined as definition 1.

Definition 1. *Similarity Relation in FAH*
The membership function on similarity relation R, μ_R: V × V → (0, 1), where R={(v_1, v_2) | v_1 and v_2 are 1-level siblings or have 1-level abstraction relationship} and V is the set of values in FAH.

Using the composition of fuzzy relation, the similarity strength between n-level siblings that are not included in the relation R can be derived from their (n-1)-level abstract values that have explicit strength. For example, the strength between a_1 and b_1 in Figure 2 can be derived through the *composition* of $\mu_R(a_1, p_1)$, $\mu_R(p_1, p_2)$, and $\mu_R(p_2, b_1)$. Generally, the *max-min composition* of two relations R and S (i.e., R•S) and is defined with the following membership function:

$$\mu_{R \bullet S}(x, z) = \mathrm{Max}_y\,[\mathrm{Min}(\mu_R(x, y), \mu_S(y, z))], \quad (1)$$
where $(x, y) \in$ R, $(y, z) \in$ S.

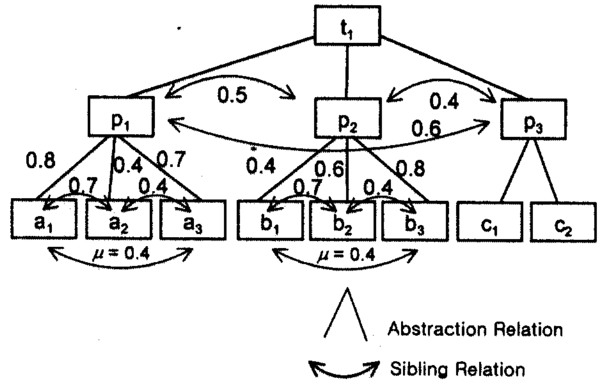

Figure 3 - Discriminative Similarity Strength in FAH.

From the definitions of the composition, it follows at once that, for any fuzzy relations R⊆X×Y, S⊆Y×Z, and T⊆Z×W, we have R•(S•T) = (R•S)•T (i.e., *associative property*). Thus, the similarity strength between 2-level siblings, a_1 and b_1, becomes

$$\mu(a_1, b_1) = \mathrm{Max}_{p_i, p_j}[\mathrm{Min}(\mu(a_1, p_i), \mu(p_i, p_j), \mu(p_j, b_1))], \quad (2)$$

where p_i and p_j are 1-level abstract values of a_1 and b_1, or 1-level siblings of the abstract values.

In (2), note that (because the abstraction strengths concerns only the values having the 1-level abstraction relationship,) the Min operator always includes 0 as $\mu(a_1, p_i)$ or $\mu(p_j, b_1)$ except for the case of ($\mu(a_1, p_1)$, $\mu(p_1, p_2)$, $\mu(p_2, b_1)$), where p_1, and p_2 are 1-level abstract values of a_1 and b_1, respectively. We call (a_1 - p_1 - p_2 - b_1) a *feasible relation path* of (a_1, b_1), which does not contain zero similarity strength between adjacent values in itself. The feasible path of a_1 and b_1 is unique in a FAH and consisted of 1-level abstract values of a_1 and b_1. Thus, (2) becomes

$$\mu(a_1, b_1) = \mathrm{Min}(\mu(a_1, p_1), \mu(p_1, p_2), \mu(p_2, b_1)), \quad (3)$$

where a_1, p_1, p_2, b_1 consist the feasible relation path of (a_1, b_1).

This composition operator can be used to derive the similarity strength of pairs that don't have explicit strength, which is called *derived strength*. However, the operator cannot guarantee to satisfy proposition 1. Thus, we need to define 'extended similarity strength' that introduces the abstraction level as follows:

Definition 2. *Extended Similarity Strength Adopting Abstraction Level*
The extended similarity strength between n-level siblings is $\lambda=(D-n)+\mu$, where D is the depth of the hierarchy and μ is the sibling strength (n=1) or derived strength ($n \geq 2$).

According to Definition 2, the similarity strength between n-

level siblings is composed of the derived similarity strength and the abstraction level. For example, the similarity strength between 'Finance' and 'OR' is λ_R(Finance, OR) = (3−2)+0.6 because they are 2-level siblings and their derived strength μ_R(Finance, OR)=0.6. Notice that the extended notion of similarity strength always satisfies proposition 1. In other words, the similarity strength between n-level siblings are always smaller than that between (n-l)-level siblings, where $n \geq 2$. Conclusively, the administrator only needs to assess the primitive strength (i.e., sibling strength and abstraction strength), while it is possible to derive the similarity strength for every pair of values with the extended similarity strength without violating proposition 1.

4. Query Relaxation Operators and Thesaurus for Query Processing

In this section, query relaxation operators are developed on the basis of the FAH and the query answering process using the thesaurus is illustrated.

4.1 Basic Operators for Query Relaxation

The semantic knowledge base executes query relaxation through value generalization, specialization, and approximation in Table 1. We can control the range of the query relaxation flexibly by adjusting the abstraction level of n-level generalization (i.e., *Generalize*(v, n)) and specialization (i.e., *Specialize*(v, n)) operations. In principle, an n-level abstract value and specific values can be obtained after recursively repeating the 1-level operations n times. To obtain n-level siblings that have derived similarity strengths greater than the given strength μ, *Approximate*(v, n, μ) executes ($n-1$)-level generalization, 1-level approximation, and ($n-1$)-level specialization.

Table 1. n-level Operators for Query Transformation

Functions	Description
Generalize(v, n)	It returns n-level abstract value of given value v and the composite strength between the abstract value and the value v by repeating the 1-level generalization n times. If n = 0, it returns v itself.
Specialize(v, n)	It returns a set of n-level specific values of given value v and the composite strength between the specific values and the value v by repeating the 1-level specialization n times. If n = 0, it returns v itself.
Approximate(v, n, μ)	It returns a set of n-level siblings of value v and the derived strength above the given strength μ. It executes ($n-1$)-level generalization, 1-level approximation, and ($n-1$)-level specialization. If the n =0, it returns v itself.

For example in Figure 1, *Approximate*(Accounting, 2, 0.7) is executed in following steps:

Approximate(Accounting, 2, 0.7)
Step 1. Generalize(Accounting, 1)=Management(0.8)
Step 2. Approximate(Management(0.8), 1, 0.7) = {Industrial Engineering(0.8)}
Step 3. Specialize(Industrial Engineering(0.8), 1)={OR(0.7), Manufacturing Systems(0.8)}
Final Result
Approximate(Accounting, 2, 0.7) = {OR((3-2)+0.7}, Manufacturing Systems((3-2)+0.8)}

The final result of *Approximate*(Accounting, 2, 0.7) is {OR, Manufacturing Systems}, and their extended similarity strength λ(Accounting, OR)=(3-2)+0.7 and λ(Accounting, Manufacturing Systems)=(3-2)+0.8.

4.2 Thesaurus

Although the query relaxation based on the FAH can treat the similarity relationship and strength between data values, it concerns only the values existing in the FAH. If users write queries with the values that do not exist in the FAH, query processing can not treat the queries just by the semantics involved in the FAH. Thus, query processing needs to be fault-tolerant to the mistakes that occur when users write the queries. In other words, users often spell incorrectly or use the allophonic words. The thesaurus for synonym information about values in the FAH can correct this problem.

Word	Token
Operations Research	OR
Macroeconomics	Macro Economics
Microeconomics	Micro Economics
Micro-Economics	Micro Economics

Figure 4 - THESAURUS Relaxation.

4.3 Query Relaxation Process

For the demonstration and explanation of the query answering processes, we use a simplified personnel database that is defined as the following:

EMPLOYEE {(<u>id</u>, emp_name, dept, title)}
COLLEGE_MAJOR{(<u>id</u>, <u>entrance_data</u>, major, graduation_date)}

The EMPLOYEE relation provides the current job position information of an employee. The COLLEGE_MAJOR relation contains the college education records.

Original Query

```
select  e.emp_name, e.dept
from    employee e, college_major c
where   c.major =? "Macroeconomics" and e.id = c.id
```

The query result of this original query written by a user is null response because 'Macroeconomics' does not exist in 'College Major' in Figure 1 and COLLEGE_MAJOR' relation. In the query, the query relaxation is specified by the relaxation operator, =?. We can replace 'Macroeconomics' with 'Macro Economics' that can be retrieved from THESAURUS relation.

Now, the generalized query at the one level is made as follows by finding 1-level abstract value of 'Macro Economics'.

Query Generalization

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

An additional query language construct, **is-a**, indicates the generalization relationship between a specialized value and an abstract value. In the generalized query, *Generalize*("Macro Economics", 1) in the generalized query returns the abstract value, Economics, and thus the query condition is relaxed as "where c.major **is-a** 'Economics' and e.id = c.id". The **is-a** operator transforms the abstract value, Economics, in the following manner, to evaluate if the c.major has membership in the set of specialized values of Economics.

Query Specialization

```
select   e.emp_name, e.dept
from     employee e, college_major c
where    c.major in Specialize("Economics", 1) and e.id =
c.id
```

The 1-level specialization of the abstract value, Economics, returns a set of specialized values {Macro Economics, Micro Economics, Econometrics} as neighborhood values around Macro Economics. Thus, the query condition of specialized query is finally written as "c.major in ('Macro Economics', 'Micro Economics', 'Econometrics') and can be answered as an ordinary SQL query.

5. Conclusion

To provide approximate neighborhood answers in addition to exact answers, we developed a knowledge representation framework FAH by combining the abstraction hierarchy and the semantic distance approaches. The FAH integrates the notion of data abstraction and fuzzy relations to represent the similarity strength between values.

As a result, FAH acquires richer semantic knowledge of an underlying database and embraces the advantages of existing data abstraction and semantic distance approaches.

First, the notion of data abstraction of FAH allows conceptually abstracted queries that are posed at a higher conceptual level based on the user's context without detailed knowledge of the database schema. Second, it assists users in the priority of each answer in the result set by providing query result sets ranked according to the similarity strength (i.e., fitness score). Third, the similarity strengths enable the users to relax the query condition more elaborately and flexibly. Furthermore, compared with semantic distance approach, FAH considerably reduces maintenance costs by decreasing the number of similarity strength to be assigned.

References

1. J. L. Braga, A. H. F. Laender and C. V. Ramos, "A Knowledge-Based Approach to Cooperative Relational Database Querying," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 14, pp. 73-90, February 2000.
2. Y. Cai, N. Cercone, and J. Han, "Attribute-Oriented Induction in Relational Databases," in *Knowledge Discovery in Databases*, AAAI Press/ The MIT Press, 1993.
3. Q. Chen, W. Chu and R. Lee, "Providing Cooperative Answers via Knowledge-Based Type Abstraction and Refinement," in *Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems*, Knoxville, Tennessee, 1990.
4. W. Chu, H. Yang, K. Chiang, M. Minock, G. Chow, and C. Larson, "CoBase: A Scalable and Extensible Cooperative Information System," *International Journal of Intelligence Information Systems*. Vol. 6, 1996.
5. W. Chu, H. Yang, and G. Chow, "A Cooperative Database System (CoBase) for Query Relaxation," in *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems*. Edinburgh, May 1996.
6. W. Chu and Q. Chen, "A Structured Approach for Cooperative Query Answering," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 5, pp. 738-749, October 1994.
7. W. Chu, Q. Chen, and R. Lee, "Cooperative Query Answering via Type Abstraction Hierarchy," in *S.M. Deen, editor, Cooperative Knowledge Base System*, pp. 271-292, North-Holland, Elservier Science Publishing Co., Inc., 1991.
8. F. Cuppens and R. Demolombe, "Cooperative Answering: A Methodologies to Provide Intelligent Access to Databases," in *Proceedings of the 2nd International Conference on Expert Database Systems*, pp. 621-643, October 1989.
9. G. J. De Sean and A. Z. Furtado, "Towards a Cooperative Question-Answering Model," *Lecture nodes in computer science*, issue 1495, pp. 354-365, 1998.
10. P. Godfrey, "Minimization in Cooperative Response to Failing Database Queries," *International Journal of Cooperative Information Systems*, Vol. 6, No. 2, pp. 95-149, 1997.
11. P. Godfrey, J. Minker, and L. Novik, "An Architecture for a Cooperative Database System," in *Proceedings of the 1994 International Conference on Applications of Databases*, June, 1994.
12. A. Hemerly, M. Casanova, and A. Furtado, "Exploiting

User Models to Avoid Misconstruals," *Nonstandard Queries and Nonstandard Answers*, Oxford Science Publications, pp. 73-98, 1994.
13. S. Huh and J. W. Lee, "Providing Approximate Answers Using a Knowledge Abstraction Database," *Journal of Database Management*, Vol.2, pp. 14-24, 2001.
14. S.-Y. Huh and K.-H. Moon, "A Data Abstraction Approach for Query Relaxation," *Information and Software Technology*, Vol. 42, pp. 407-418, 2000.
15. W. S. Li and D. Agrawal, "Supporting Web Query Expansion Efficiently Using Multi-granularity Indexing and Query Processing," *Data & Knowledge Engineering*, Vol. 35, pp. 239-257, December 2000.
16. M. J. Minock and W. Chu, "Explanation for Cooperative Information Systems," in *Proc. of Ninth International Symposium on Methodologies for Intelligent Systems*. June 1996.
17. A. Motro, "VAGUE: A User Interface to Relational Databases that Permits Vague Queries," *ACM Transactions on Office Information Systems*, Vol. 6, No. 3, pp. 187-214, July 1988.
18. A. Motro, "FLEX: A Tolerent and Cooperative User Interface to Databases," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 2, pp. 231-246, June 1990.
19. A. Motro, "Intensional Answers to Database Queries," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 3, pp. 444-454, June 1994.
20. D. C. C. Poo, T, K. Toh, and C. S. G Khoo, and G Hong, "Development of an intelligent Web interface to online library catalog databases," Software Engineering Conference, 1999. (APSEC '99) Proceedings. Sixth Asia Pacific, pp. 64 -71, 1999.
21. S. Ram, "Intelligent Database Design Using the Unifying Semantic Model," *Information and Management*, Vol. 29, No. 4, pp. 191-206, 1995.
22. S. V. Vrbsky and W. S. Liu, "APPROXIMATE-A Query Processor that Produces Monotonically Improving Approximate Answers," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 6, pp. 1056-1068, 1993.
23. M. Zemankova and Kandel, A., "Implementing Imprecision in Information Systems," Information Science, Vol. 37, No. 1, Dec. 1985, pp 107-141