Applying Metricized Knowledge Abstraction Hierarchy for Securely Personalized Context-Aware Cooperative Query

Ohbyung Kwon [a], Myunggeun Shin [b] and Injun Kim [c]**

[a] *School of International Management, Kyunghee University*
1, Seochun-Dong, Ghyheung-Gu, Yongin, 449-701, Korea
Tel: +82-31-201-2306, Fax: +82-31-204-8113, E-mail: obkwon@khu.ac.kr

[b] *2EIT Inc.*
17th Mesa Building, Hoihyung-Dong, Jung-Gu, Seould, Korea
Tel: +82 2 2128 7004, Fax: +82 2 2128 7036, E-mail: mkshin98@kaist.ac.kr

[c] *Department of Computer Engineering, Kyunghee University*
1, Seochun-Dong, Ghyheung-Gu, Yongin, 449-701, Korea
Tel: +82-31-201-2566, Fax: +82-31-202-1723, E-mail: foel064@khu.ac.kr

Abstract

The purpose of this paper is to propose a securely personalized context-aware cooperative query that supports a multi-level data abstraction hierarchy and conceptual distance metric among data values, while considering privacy concerns around user context awareness. The conceptual distance expresses a semantic similarity among data values with a quantitative measure, and thus the conceptual distance enables query results to be ranked. To show the feasibility of the methodology proposed in this paper, we have implemented a prototype system in the area of site search in a large-scale shopping mall.

Keywords:

Context-aware Computing; Cooperative Query; Abstraction Hierarchy; Approximate Query

Introduction

The purpose of context-aware computing is generally to make computer systems into a more natural part of peoples' everyday life by making the technology more flexible and attentive. So to speak, context-aware computing is to make computing fit to peoples' ordinary lives and the activities they devote themselves to. With the help of context-aware computing, context-aware services aim at adapting responses to eventual context related changes in an unobtrusive manner. Unobtrusiveness is a key philosophy that discerns between legacy personalization systems and context-aware systems. Moreover, unobtrusiveness is tightly linked to privacy concerns, which is one of the main concerns of using context-aware systems, in that users frequently feel uncomfortable when the system compels them to input personal data for more services, or is even suspected of collecting and reusing users' data without the users' permission or approval. At a minimum, this requires an increased awareness and sensitivity to users' internal and external surroundings in order to be aware of the users' behavior intention.

To achieve this unobtrusiveness, several conditions must be met. First, the context-aware services should ask for only the minimum user input possible. In particular, when we assume that the more sophisticated context-based systems would also need more context data, we cannot imagine that the user will accept such systems unless the systems automatically acquire context data. Accordingly, if the context-aware system encompasses a query system, then the query commands should be made with minimal user input. Second, the context-aware system should resolve the user's privacy concerns around having to input personal data to enable further services. Third, a more realistic scenario allows users to submit vague queries rather than exact matches. Vague queries are common when the users do not actually know what they want. To address the unobtrusive queries in a context-aware system, accurately resolving vague queries in a way that takes both context-awareness and privacy concerns must be considered.

Researchers have proposed the cooperative query as a way to enable users' vague queries. The essence of the cooperative query for vague query approach is to compute the concept distance. However, legacy cooperative query algorithms fail to reflect how the changes of the user's current context affect concept distance computation. Moreover, current cooperative queries do not consider privacy concerns. These limitations naturally lead us to developing an unobtrusive context-aware cooperative query system for a personalized recommendation.

Hence, the purpose of this paper is to propose a securely personalized context-aware query methodology based on context data such as user's current location and schedule. The methodology resolves users' vague queries and personal preferences around privacy concerns by requesting only minimal input for query processing. To show the feasibility of the methodology proposed in this paper, a prototype system, CACO (Context-Aware COoperative Query system), was implemented and actually used at COEX, one of the largest shopping malls in Korea.

Literature Review: Cooperative Query

Query languages have been widely used as convenient tools to retrieve information from databases. Most conventional database systems support specific queries for only data that match precise query qualifications. Therefore, database users have to fully understand both the metadata and contents of the database. Even the users who are familiar to the database have to retry specific queries repeatedly with alternative values until the query result is satisfactory. However, if the exact answer for the queries does not exist, the system would provide limited answers or even no information at all.

To remedy such restrictions, cooperative query answering (Barg and Wong 2000, Bernardno et al. 2002, Bosc et al. 2001, Gaasterland 1997, Han et al. 1996, Babcock et al. 2003, Liu and Ng 1998, Wang et al. 2004) was proposed, which provides relevant information of a wider scope or even an approximate answer. Typical steps for cooperative query answering include query analysis, query relaxation, and providing information relevant to or associated with the query. To facilitate query relaxation and the approximate or associated information search, testers use a knowledge representation framework, which is one of the most important factors that can affect overall performance and the characteristics of the cooperative query answering system.

A large portion of the studies have been done with the logic model approach (Braga et al. 2000, De Sean and Furtado 1998, Gaasterland 1997, Godfrey 1997). The logic model approach 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 inquired information is specified with free variables. The query is answered through conflict resolution and an inference mechanism. Query relaxation is effected by coordinating the integrity constraints. However, this approach has limitations in guiding both the query relaxation process and the less intuitive query answering process. Therefore, the logic model approach is not adequate to build up a large scale system.

The abstraction approach (Chu et al. 1996, Huh and Lee 2001, Vrbsky and Liu 1993) uses the data abstraction method that has been considered effective to accommodate the semantic relationships among data values (Abiteboul and Duschka 1998, Abiteboul et al. 2002). In cooperative query answering, such data abstraction is also useful in associating data values for query relaxation. To validate the abstract concepts that comprise the related data values, the type abstraction hierarchy (Chu and Chun 1994) introduced the notions of subsumption, composition, and abstraction, and offered an integrated view of the type hierarchy with a multi-level knowledge abstraction. The type abstraction hierarchy used both query rewrite and subject association to facilitate reasoning among different knowledge levels in order to derive cooperative answers. The knowledge abstraction hierarchy (KAH) extended the type abstraction hierarchy, and it focused capturing value abstraction information with additional abstraction information that represented domain abstraction knowledge elicited from the underlying databases (Huh and Lee 2001). However, these methods do not provide a 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. Moreover, this approach is not appropriate to build up a scalable and extendable cooperative query answering system. In brief, adopting the semantic distance notion to the abstraction hierarchy would overcome the weaknesses of the approaches, and provides a more effective and extendable cooperative query answering mechanism that can support a wider range of approximate queries.

The semantic distance approach (Cuzzocrea and Matrangolo 2004, Ichikawa and Hirakawa 1986, Motro 1988, Motro 1996, Palpanas 2001) 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 (Motro 1988), and thus this approach provides a 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 the semantic distance does not consider the user's context, dynamic and personalized query processing is not doable.

Illustrative Example

To explore and select an appropriate query method adequate to build up a large scaled system which is actually running, we selected the Korean COEX mall in this paper. To make shopping easier, the mall's walkways are 18 meters wide (the widest in Korea), and up to 663 meters long. The number of visitors is approximately 150,000 on weekdays and 250,000 on weekends. Moreover, a number of adjacent buildings and spaces, such as department stores, hotels, subway, City Air Terminal, and towers, give a synergistic effect. In particular, COEX mall has a number of sophisticated loyalty programs targeted towards customers called "mallzok," or "people living in COEX mall." These people spend a great deal of their non-work hours at COEX, including meeting, dating, and entertainment.

However, the COEX mall still has room for improvement vis à via searching shops. First, the users tend to visit COEX mall not determining themselves where to go a priori, but starting to search for an appropriate place according to the visiting purposes. In other words, a number of visitors would benefit from a purpose-driven, ad-hoc search at a point of need, as well as spot-based search. Moreover, COEX is strongly interested in attracting the hundreds of thousands of people visiting the adjacent attractions such as the convention center, hotel, or nearby business buildings. However, COEX's CEO notes that most of the people visiting peripheral attractions are not promoted well, and hence just pass through the mall. Current information desks for mall visitors do not provide a purpose-driven ad-hoc search. Second, like most malls, COEX provides visitors with only a predefined categories of the tenant shops (men's clothing, women's clothing, specialty, etc.) The categorizing rules need to be personalized according to the users, since often it is difficult to find the right place by seeing the predefined category. Third, the current support system is vulnerable to when shops change. From time to time, locations of the spots shown in the map are not identical to the actual locations due to new or changed mall entries. Especially in these cases, even small changes in the shopping information can create considerable visitor confusion.

Securely Personalized and Context-Aware Cooperative Query

Notion

Traditional queries, represented as conventional SQL commands, typically support an exact query. In comparison with traditional queries, a cooperative query is innovative in that it can provide a wider scope of relevant information, or even an approximate answer. However, cooperative queries themselves are not aware of user context, and hence support only static queries.

On the other hand, a context-aware query is motivated from the fact that queries processed on the World Wide Web frequently do not return desired results because they fail to take into account the context of the query and information about users situation and preferences. In Storey et al.'s approach (2004), user profiles were proposed as a way to increase the accuracy of web pages returned from the Web. NetTraveler addressed some research problems on context-aware query under web services, XML-based profiles, and Peer-to-Peer search protocols, with the aid of the ability of tracking mobile devices (Caituiro-Monge and Rodriguez-Martinez, 2004). However, the context-aware query methods so far have not considered the vague query. This limitation could be serious so that the query system may be actually utilized mainly because the context data are hard to be asserted in a unified manner. Therefore, if we conjointly use a context-aware query and a cooperative query, perhaps a context-aware cooperative query, then real-time vague queries would be viable.

A personalized context-aware cooperative query is a context-aware cooperative query that identifies the user's preference data, which is stored somewhere that is accessible by the query system. To accomplish this, the personal preference manager might use web pages and announce the URL or URI of the pages to be used by external programs such as query systems and agents. The preference data enables the context-aware cooperative query system to be more personalized. However, this kind of query system might be obtrusive to the user, in that the query may require collecting context data around which the user has some privacy concerns. Therefore, the proposed personalized context-aware cooperative query system must be securely managed. This personalization might increase the correctness of the query results yet guarantee unobtrusive use of query system. Consequently, the methodology of this paper tries to realize the securely personalized context-aware cooperative query.

Concept Distance Metric and Its Calculation in the Hierarchy of Static MKAH

In this section we represent the concept of the distance metric and definition of the shortest pass, and we define the distance which satisfies the requirement of the concept distance metric.

First of all, the concept distance in the hierarchy of static MKAH is acquired through the following steps.

Step 1: Identify value abstraction hierarchy, which is the former MKAH.

In our COEX example, the value abstraction hierarchy of the sites are shown as Figure 1.

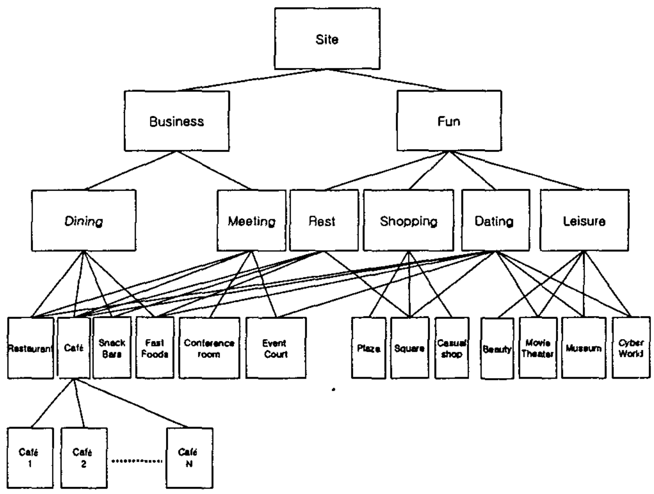

Figure 1 - Example of value abstraction hierarchy

Step 2: Identify a set of upcoming activities.

In our example, six activities are predefined: rest, meeting, dining, dating, shopping, and leisure.

Step 3: Assert contextual concept distance of each node as follow:

cd(node_value, ACTIVITY = activity_value).

Even though the concept distance is set to a default value by the system, the user can subjectively and personally determine that value. Or, if the user prefers not to use personal data and is also not satisfied with the default value, the system may adopt the concept distance value to the user's preference value with history data. In this paper, case based reasoning is used to provide adaptive estimation concept distance values.

Step 4: With the use of concept distance declared in the Step 3, compute the static concept distance between any two nodes. The static concept distance is acquired by the following function if the nodes are president node and vice president node:

$$cd(nodeA, nodeB) = (d_{y,1} + d_{y,2} + ... + d_{y,N}) / N$$

where nodeA and nodeB are the president node and vice president node, respectively.

N indicates the total number of activities.

If the nodes are neither president nor vice-president, then the following function is applied:

$$cd(nodeA, nodeB) = \sqrt{(d_{x,1} - d_{y,1})^2 + (d_{x,2} - d_{y,2})^2 + ... + (d_{x,N} - d_{y,N})^2}$$

For example, in case of the two nodes indicated restaurant and square, the static concept distance is:

$$cd(Restaurants, Square) = \sqrt{(0.6-0.3)^2 + (0.4-0.2)^2 + ... + (0.8-0.7)^2} = 0.435$$

Step 5: Using the static concept distance derived in the Step 4, get static MKAH.

Contextual MKAH

Getting contextual MKAH starts from the static MKAH of the Step 5.

Step 6: Using the static concept distance derived from the Step 5, acquire the contextual concept distance of each activity. The function cd(nodeA, nodeB | Activity = "a") is a Euclidean distance of any two nodes, cd(nodeA | Activity = "a") and cd(nodeB | Activity = "a") when the activity is identical to each other as shown in Figure 2.

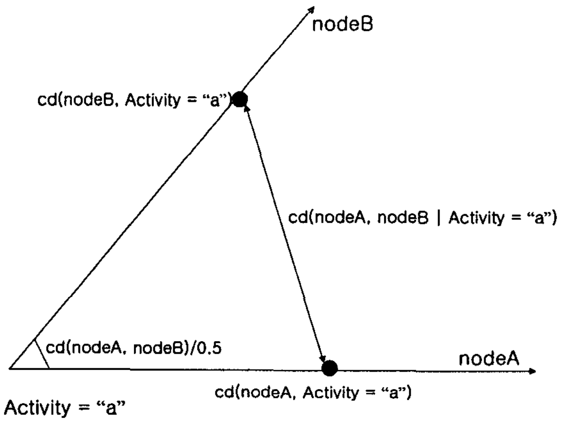

Figure 2 - Contextual concept distance

Step 7: With the contextual concept distance acquire in the previous step, identify contextual MKAH of each activity. Therefore, the number of contextual MKAH is identical to the number of activities declared in the Step 2.

Step 8: Using the contextual MKAH derived in Step 7, a securely personalized cooperative query is performed.

Step 9: To get the personalized contextual preference for each candidate node, transform a node instance, , of node A into (x,y) location by performing a transforming function f:

$$\alpha \xrightarrow{f} (x, y)$$

The (x,y) location of each node is used to compute the physical distance from the user's current location to the node location. In this paper, Euclidean distance is used.

Step 10: If the system acquires PSCP on distance through the user interface, then for any node perceived distance (pd) is estimated as:

$$pd(\alpha) = PhysicalDistance^{g(p)}$$

where and p indicates a function of PSCP and degree of PSCP ranged from 1 to 5, respectively.

Step 11: Finally, personal preference in terms of concept distance on each node is adjusted using the context of perceived distance. For $\forall \alpha \in nodeA$,

Personal preference on $\alpha = cd(nodeA, nodeB) * pd(\alpha)$

CACO: A Prototype System

Based on the securely personalized context-aware query method proposed in this paper, a prototype system, CACO (Context-Aware COoperative Query system), was developed and tested at the COEX mall in Seoul, Korea. CACO is a ubiquitous services platform that provides flexible node recommendations under given conditions in terms of user profile, context, and privacy concerns preferences. CACO can automatically identify user context, such as the user's current location and upcoming activities. Multi-agent architecture is considered in CACO, providing both a UA (User Agent) and a TSA (Task Specific Agent). The agent systems are implemented with JDK 1.4.x. A web-based user interface is implemented with JSP. The user and service ontologies that CACO uses are represented with OWL and Jena. To support the distance calculation between user's current location and a specific service node, a web service called DistanceService is implemented as an Apache SOAP 2.0 Server. We used Jet Engine as the case base and database in CACO. The proposed system architecture of CACO is shown as Figure 3.

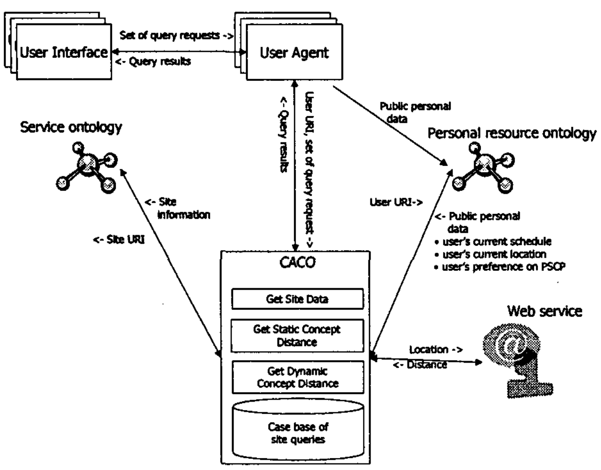

Figure 3 - CACO System Architecture

An example interface of CACO is shown as Figure 4. As seen in lower left window of Figure 4, CACO allows the user to put four kinds of data: node, the user's PSCP (perceived sensitivity of context pressure) on distance, upcoming activity, and mode of match. The total number of nodes which are derived from the example domain is 15 according to the COEX mall domain. Additionally, CACO allows the user to select "automatic," if the user does not determine where to go even though the user wants to go somewhere. As noted earlier, the perceived sensitivity on contextual pressure concept is rooted in psychology's idea of stress and neuroticism (Schuler, 1980; Lazarus, 1984; Gonzales et al., 2001). In the setting of using a context-aware service system such as CACO, users will behave differently due to the extent to which an individual depresses under contextual pressure. With increased contextual pressure, a person may produce poorer performance. Adopting this model to the CACO implementation, looking especially at the distance from the user's current location to a site's location, those with high-neuroticism levels are expected to have a higher anxiety when using a new information system, because of their personality, in terms of going to a site which is located relatively more remotely.

In CACO, the level of PSCP on distance is acquired from the user with 5-Likert scale. "Upcoming activity" is that which the user is about to perform in a site. The activities are classified into three categories: user's manual input, referring user's ontology on daily schedule, and privacy mode. By manual input, the user can assert her or his upcoming activity to CACO. If the user selects "as my schedule," then CACO visits the user ontology, imports the user's daily schedule, and identifies the upcoming activity using a simple query command with the condition of current time. The user may not want to disclose his or her upcoming activity because of privacy concerns. In this case, CACO accomodates the user's privacy preferences by having the user select "privacy." Matching modes are two-fold: exact match and ambiguous match. Exact match is to find sites which are exactly identical to the node that the user selects. Ambiguous match allows CACO to search another type of sites using cooperative query proposed in this paper.

Using the four parameters described, CACO shows a personalized and context-awarely considered query results. For example, suppose the user wants to take a rest at a certain site in COEX mall and her/his PSCP on distance is scaled as 3, 'so so', at 10:00am. The user may select the parameters as follows:

Case 1: *get recommendations from CACO by exact match*

Case 2: get recommendations from CACO by ambiguous match

Case 2: get recommendations from CACO by ambiguous match and accepting privacy concern

The results of cases 1, 2, and 3, respectively, are shown as upper-left, upper-right, and lower-right windows of Figure 4. The results are quite interesting and meaningful. As shown in the upper-left window, searching restaurants using exact match and sorting them by distance displays four restaurants, which are all enrolled in the service ontology. According to the distance, 'jugjug_beer' is selected as top recommendation. The number 0.23 indicates the penalty of the sites in terms of location-based concept similarity. In case 2, CACO performs ambiguous match, which means to recommend sites according to the concept similarity of "dating" places, rather than considering simply "restaurant," as the user selects in the initial page. In this case, CACO realizes more conceptually appropriate sites other than restaurants: fastfoods, square, snack bars, etc. In case 3, the privacy mode, even though CACO does not aquire any explicit information from the user othern than that the user wants to go to a restaurant, CACO flexibly suggests more site candidates other than just restaurants using a concept similarity vector. This implies that CACO overcomes the 'mesa effect', which is one of the representative limitations of the legacy query systems and rule-based expert systems with static rules.

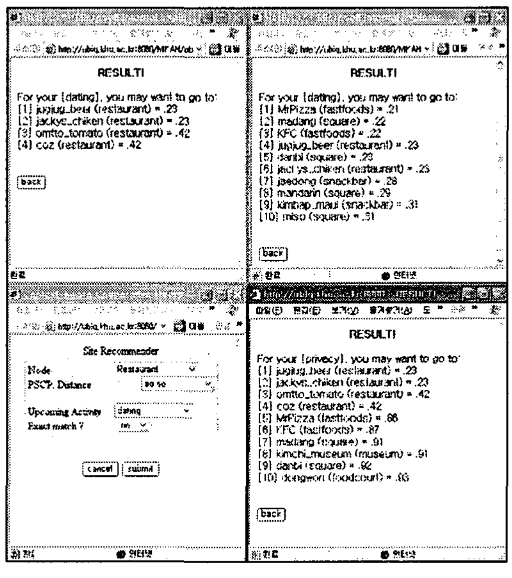

Figure 4 - CACO Example Screenshots about mode of match and activity

Figure 5 show an actual interface using PDA with RFID reader and wireless LAN card.

Figure 5 - Actual User Interface

Conclusion

In this paper, we propose a securely personalized context-aware cooperative query method using dynamic concept similarity, and we show how this method can be applied to context-aware personalization services. In particular, recent mobile and ubiquitous computing systems make it possible to fully make use of a user's context, such current location and schedule. Among those, a recommendation service is promising as a context-aware application domain that is acceptable to nomadic users. The core of the recommendation service success is a personalized query method. However, cooperative query systems that consider context-awareness for personalization are very few. Moreover, legacy cooperative query systems do not work well when the user has privacy concerns in terms of revealing his or her schedule to unknown or unauthorized recommendation systems. Hence, our approach addresses these limitations by applying dynamic concept similarity to the cooperative query method. To show the feasibility of the idea proposed in this paper, a prototype system, CACO, was developed and tested on actual users' wireless devices.

Acknowledgements

This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

References

[1] Abiteboul, S. and Duschka, O. M. (1998). "Complexity of Answering Queries Using Materialized Views," *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Seattle, Washington, pp. 254-263.

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

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

[4] Barg, M. and Wong, R. K. (2000). "A Multi-Agent Architecture for Cooperative Query Answering", *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Maui, Hawaii.

[5] Bernardino, J. Furtado, P. and Madeira, H. (2002) "Approximate Query Answering Using Data Warehouse Striping," *Journal of Intelligent Information Systems*, Vol. 19, No. 2, pp. 145-167.

[6] Bosc, P., Motro, A. and Pasi, G. (2001). "Report on The fourth International Conference on Flexible Query Answering systems," *SIGMOD Record*, Vol. 30, No. 1, pp. 66-69.

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

[8] Caituiro-Monge, H., Rodriguez-Martinez, M., Nettraveler. (2004). A Framework For Autonomic Webservices Collaboration: Orchestration And Choreography in e-government Information Systems, *Proceedings of the IEEE International Conference on WebServices*, pp. 2-9.

[9] Chu, W., Yang, H. and Chow, G. (1996). "A Cooperative Database System (CoBase) for Query Relaxation," *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems*, Edinburgh.

[10] Cuzzocrea, A. and Matrangolo, U. (2004). "Analytical Synopses for Approximate Query Answering in OLAP Environments," *Proceedings of the 15rd International Conference on Database and expert systems applications*, Zaragoza, Spain, pp. 359-370.

[11] De Sean, G. J. and Furtado, A. Z.. (1998). "Towards a Cooperative Question-Answering Model," *Lecture notes in computer science*, Vol. 1495, pp. 354-365.

[12] Gaasterland, T. (1997). "Cooperative Answering through Controlled Query Relaxation," *IEEE Expert*, Vol. 12, No. 5, pp. 48-59.

[13] Godfrey, P. (1997). "Minimization in Cooperative Response to Failing Database Queries," *International Journal of Cooperative Information Systems*, Vol. 6, No. 2, pp. 95-149.

[14] Gonzales, N.A., Tein, J., Sandler, I.N., and Friedman, R.J. (2001). "On the limits of coping: interaction between stress and coping for inner-city adolescents,"

Journal of Adolescent Research, 16, 4, pp. 372-395.

[15] Han, J., Huang, Y. and Cercone, N. (1996). "Intelligent Query Answering by Knowledge Discovery Techniques," *IEEE transactions on knowledge and data engineering*, Vol. 8, No. 3, pp. 373-390.

[16] Hou, W. (1996). "Extraction and Applications of Statistical Relationships in Relational Databases," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, pp. 939-945.

[17] Ichikawa, T. and Hirakawa, M. (1986). "ARES: A Relational Database with the Capability of Performing Flexible Interpretation of Queries," *IEEE Transaction on Software Engineering*, Vol. 12, No. 5, pp. 624-634.

[18] Lazarus, R. and Folkman, S. (1984). *Stress, appraisal, and Coping*, New York: Springer.

[19] Liu, Q. and Ng, P. A.. (1998). "A Query Generalizer for Providing Cooperative Responses in an Office Document System," *Data and Knowledge Engineering*, Vol. 27, No. 2, pp. 177-205.

[20] Motro, A. (1988). "VAGUE: A User Interface to Relational Databases that Permits Vague Queries," *ACM Transactions on Office Information Systems*, Vol. 6, No. 3, pp. 187-214..

[21] Motro, A. (1996). "Cooperative Database Systems," *International Journal of Intelligent Systems*, Vol. 11, No. 10, pp. 717-732.

[22] Palpanas, T. and Koudas, N. (2001). "Entropy Based Approximate Querying and Exploration of Datacubes," *Proceedings of the Scientific and Statistical DataBase Management*, Fairfax, VA, USA, pp.81-90.

[23] Schuler, R. (1980). Definition and Conceptualization of Stress in *Organizations. Organizational Behavior and Human Performance*, 25, pp. 184-215.

[24] Storey, V.C., Sugumaran, V. and Burton-Jones, A. (2004). "The Role of User Profiles in Context-Aware Query Processing for the Semantic Web," *Lecture Notes in Computer Science*, (Eds.) Farid Meziane, Elisabeth Metais Springer-Verlag GmbH, Vol. 3136, pp. 51-63.

[25] 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*, Vol. 5, No. 6, pp. 1056-1068.

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

C 발표장(제4소회의실) C3. e-Business I

2006 한국지능정보시스템학회 춘계학술대회

C3.1 효과적인 기업용 S/W 판매전략 공유를 위한 인지지도 기반의 암묵지 관리 방법
정남호 (충주대학교), 이남호 (성균관대학교), 이건창 (성균관대학교)

C3.2 ASP 기반 회계정보시스템 사용의도 영향요인 연구
김병섭 (한성대학교), 안병석 (한성대학교)

C3.3 e-Commerce를 위한 게임이론 기반의 지능모델
정재헌 (한국과학기술연구원), 염기원 (한국과학기술연구원),
박지형 (한국과학기술연구원)