TV Program Searching and Ranking for Supporting TV Personalization

Jungmin Kim[1], Hyunsook Chung[2] and Gukboh Kim[1]

[1]Department of Computer Engineering, Daejin University, Pocheon, Korea
[2]Department of Computer Engineering, Chosun University, Gwangju, Korea
[1]{jmkim, kgb}@daejin.ac.kr, [2]hsch@chosun.ac.kr

Abstract

Current, so many TV program contents are being provided to viewers on various channels through internet protocol televisions called IPTVs. Therefore the development of an effective searching scheme to enable viewers find and consume their preferred contents has been being important issue. This paper proposes a TV program contents searching scheme based on contents ontologies, viewer category, and usage examination. Our scheme search TV programs by computing the similarity between contents ontologies, filter the candidates with preferences of viewers, and return the list of top-N ranked program contents. As results of experiments, we know that our scheme has about 80% precision.

Keywords: *IPTV, ontology, TV program ontology, Content-based searching, ranking*

1. Introduction

Currently, IPTV has been popular more and more because it enables users to selectively consume contents of live broadcasting channels, time-shifted programs, and video on demand without time limitation. However, so many TV program contents are being provided to viewers, and thus it may cause that viewers have difficulty in finding and consuming their preferred contents [1, 2].

There are two approaches to solve this problem, which are searching contents and recommending contents. Searching contents technique means pull method because viewers determine selection of contents by entering some keywords into search engines in contents providers, whereas recommending contents technique means push method because contents providers transmit the recommended list of contents to viewers.

PTV[3] is an earlier research work on personalized TV guides development. It proposes a client-server system in which TV program recommendation is performed by executing content-based and collaborative recommendation approaches on domain preferences and program preferences of users. Zhang and Zheng [4] proposes a personalized TV system based on TV-Anytime metadata model. They creates a user profile based on CreationPreferences DS (description scheme), UsageHistoryDS, ClassificationPreferenceDS of TV-Anytime metadata and adopts the content-based filtering to recommend programs that have high similarity to user profile.

PersonalTVware [5] is an software infrastructure to support the context-aware recommendation of TV programs. This software provides TV program filtering, contextual information management, cross-context reasoning to infer contextual preferences. Kim *et. al.*[6] proposes a collaborative filtering-based recommendation system of TV programs, which consists of user profile reasoning, user clustering and TV program recommendation. Their recommendation method consists of three steps like extracting similar preference users from the similar user groups, selecting candidate programs for recommendation, and producing the ranked candidate programs.

In this paper, we consider search problem of IPTV and develop an effective TV program contents searching scheme based on TV program contents ontologies and observation of usage logs of viewers. Our approach consists of four phases that are constructing TV program contents ontologies, computing the similarity between contents ontologies, filtering the retrieved candidates with various preferences of viewers, and ranking the filtered candidates according to their rank scores.

There are a few studies related to the development of ontology search engines in order to support knowledge reuse. OntoSearch [7] is ontology search engine using Google APIs and hierarchy visualization technique. It allows users to search certain types of ontology files by keywords only. OntoSelect [8] is a web-based ontology library that collects, analyzes, and organizes ontologies and allows searching as well as browsing of ontologies according to size, representation format, connectedness and human languages used for class labels. OntoSelect provides ontology search based on one or more keywords and a HTML document.

Swoogle [9] is a crawler-based indexing and retrieving system for RDF or OWL documents. OntoKhoj [10] is a Semantic Web portal designed to simplify ontology engineering process. It is based on algorithms used for searching, aggregating, ranking and classifying ontologies in Semantic Web. OntoKhoj crawler fetches the RDF documents according to the physical links and then aggregate several RDF segments belong to same logical URI but physically present at different locations into a single ontology.

As depicted in Figure 1, our scheme accepts an ontology of consumed (watched) TV program as query data rather than simple keywords to find other relevant broadcasting program content ontologies, which have higher similarity in terms of syntactic and semantic structure, from the collection of TV program ontologies.

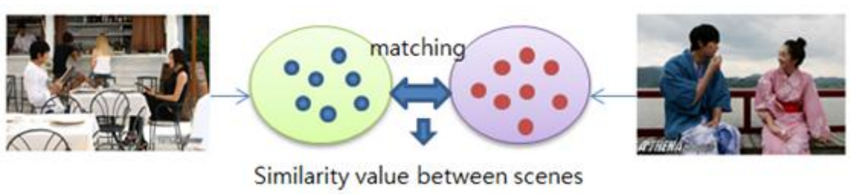

Figure 1. Comparison between Contents Ontologies to Search TV Programs

2. TV Program Contents Ontology Definition

Our TV program ontology has three-layered structure composed of schema ontology, program metadata ontology and program content ontology. Schema ontology includes high-level concepts, such as person, event, place, etc. Transforming TV-Anytime metadata schema to OWL creates program metadata ontology. Program content ontology defines concepts and relations included in TV program contents. Program content ontology is a set of independent ontologies, which conceptualize contents of broadcasting programs according to their genres like drama, news, sports, documentary, etc.

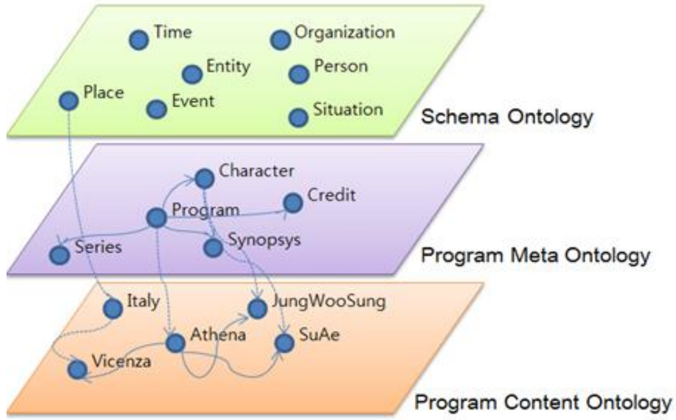

Figure 2. TV Program Ontology Architecture

Program content ontology, *i.e. Drama ontology*, *Sports ontology*, and *Documentary ontology*, conceptualizes the contents of TV programs by means of core concept identification, term definition, and semantic relation creation. Program content ontology enables users to search certain broadcasting programs or related contents to watching program. Broadcasting programs delivered from IPTV have web pages, which describe synopsis, character, credit, and so on. This basic information may be provided to users using TV-Anytime metadata. We use the auxiliary information of programs in order to extract knowledge of contents of programs.

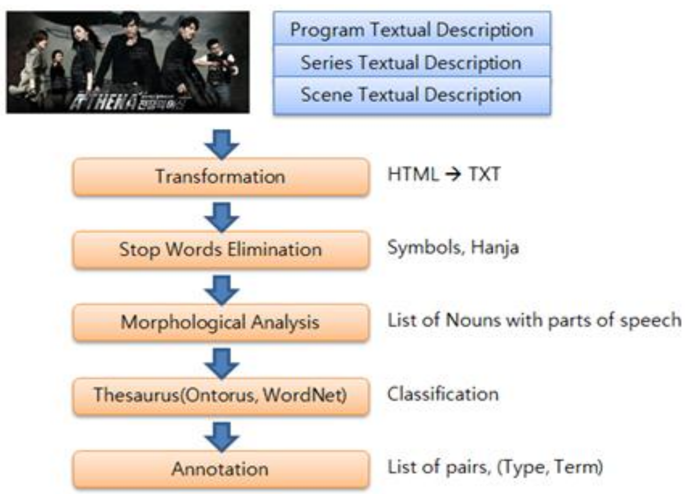

Figure 3. TV Program Content Ontology Definition Process

In Figure 3, program content ontology creation is performed as follows: First transformation step is executed to parse the collected web pages, which are textual resources related to TV programs, in order to remove unnecessary data like images, symbols, and numbers, and extract textual data. Textual data extracted from several web pages are merged in a text file and passed into morphological analyzer to identify actual morphemes. Domain experts examine actual morphemes manually in order to identify core concepts of programs. This work is processed in morpheme clean step. Next step creates the semantic relation of core concepts and instances of the concepts.

3. The Proposed Searching Scheme

In this section, we describe our TV programs searching scheme. The system architecture is represented in Figure 4.

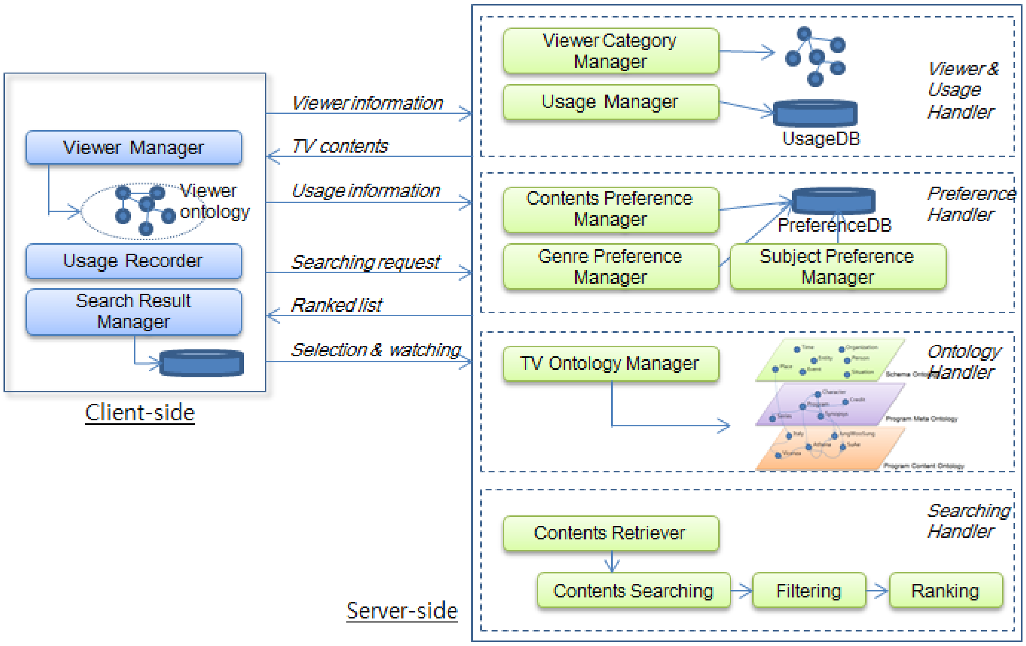

Figure 4. The System Architecture for Searching and Ranking Broadcasting Ontologies

3.1. Viewer Preference Modeling

The characteristic of viewers can be identified by examination of two kinds of data, one is demographic information and the other is TV program contents usage history. We collect gender, age, hobbies, and favorite genres of him/her as demographic information for each viewer. Collected usage history is examined to determine program preference, genre preference, and subject preference for each viewer.

$$p_u(x) = \begin{cases} 1, & \dfrac{\sum_{i=1}^{n} R_{w^i}(x)}{R(x)} \geq \alpha \\ \dfrac{\sum_{i=1}^{n} R_{w^i}(x)}{R(x)} \div \alpha, & \dfrac{\sum_{i=1}^{n} R_{w^i}(x)}{R(x)} < \alpha \end{cases} \qquad (1)$$

Expression (1) computes a program preference on TV program x consumed by viewer u. $p_u(x)$ has a floating point value between 0 and 1 result from applying normalization coefficient a. $R(x)$ denotes the running time of TV program x. $\sum_{i=1}^{n} R_{w^i}(x)$ denotes the sum of watched time for TV program x by viewer u. Genre preference can be computed by summation of program preferences belong to each genre. We use the TV-anytime genre taxonomy as standard TV program genre classification, which defines 14 genres and 494 subgenres, such as Drama, Movie, Music, Arts, etc. Subject preference represents the preferred keywords of viewer *u*. We define the subject of a certain program *x* in terms of the list of top 5 most frequent concept terms represented in its program content ontology. Subject preference can be computed by summation of program preferences for each representative keyword.

3.2. Similarity Computation between Program Ontologies

To perform similarity computation between TV program content ontologies, first, we transform program ontologies into documents, and measure how many two documents include common concept labels by using term vectors extracted from documents. As depicted in Figure 5, our TV program searching steps are processed as follows:

1) Creating term vectors of two documents(program ontologies). For a program content ontology, we create three kinds of term vectors according to the types, such as class, property, and relation, of extracted strings.

2) Computing similarity values between term vectors. For each of the entity types, term vectors belong to same type are compared based on the cosine measure, and take a similarity value as their score.

3) Combining similarity values. Similarity values should be combined to make a representative score for two program ontologies. A combined similarity score is normalized as a value from 0 to 1.

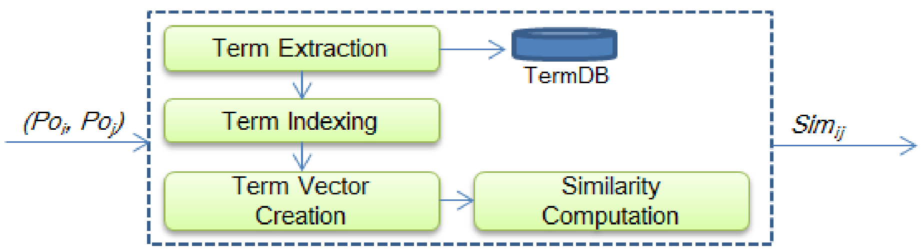

Figure 5. Similarity Computation between Program Contents Ontologies

Conventional IR engines employ a tf*idf model with a standard cosine similarity metric. In the field of Information Retrieval, the classic term vector model is based on the following expression (2). IR systems assign weights to terms by considering local information extracted from the individual document, term frequency, and global information extracted from document collection.

$$\text{Term weight } w(x) = tf(x) * \ln\left(\frac{D}{df(x)}\right) \tag{2}$$

For ontology-based TV program searching, we use the normalized frequency-based vector model instead of the classic tf*idf vector model in order to compute the similarity between TV program content ontologies because the number of ontologies containing common terms is not an important factor in the ontology search process. The normalized frequency of the term *x*, which is a weight of term *x*, in a ontology is given by the following expression:

$$w(x) = ((tf(x))/\max \llbracket tf(y) \rrbracket\ ) \tag{3}$$

$$sim(po_q, po_j) = \frac{po_q \cdot po_j}{\|po_q\|\|po_j\|} = \frac{\sum_{i=1}^{n} w_{q,i} w_{j,i}}{\sqrt{\sum_{i=1}^{n} w_{q,i}^2}\sqrt{\sum_{i=1}^{n} w_{j,i}^2}} \tag{4}$$

From the above expression, max *tf(y)* denotes the maximum frequency of the term *y*. Thus, the expression computes a term ratio for each term in a program content ontology. We create a term vector based on the normalized frequencies of the terms. Two term vectors of two ontologies are compared to compute a similarity value between them. Comparison of two term vectors is performed through computing the cosine value of two term vectors. These cosine values represent the similarity values of ontologies, and are used to sort search results in descending order before passing into the ranking module.

3.3. Filtering and Ranking the Candidate Program List

Searching process finds the conceptually overlapped TV programs based on term-level similarity. As next step, we perform filtering retrieved candidate programs with preferences of viewer. Candidate programs having group preference and subject preference below a particular threshold on are filtered out. Our ranking module takes the list of filtered program ontologies and ranks them according to their scores produced by applying three different measures, such as concept completeness, relation complexity and concept density, which estimate the quality of knowledge structures of program ontologies.

Table 1. Ranking Operations and Expressions

Operation	Expression	
Concept completeness	$tc(i) = w*\lvert ICi\rvert + (1-w)*\lvert ECi\rvert$	(5)
	$TC = \sum_{i=1}^{n} tc(i)$	(6)
Relation complexity	$trc(i) = w1*Super(i)+w2*Sub(i)+w3*I(i)+w4*A(i)+w5*sib(i)$	(7)
	$TRC = \sum_{i=1}^{n} trc(i)$	(8)
Density	$dist(t_i, t_j) = \begin{cases} length(t_i \rightarrow t_j), & if\ t_i \neq t_j \\ 1, & if\ t_i = t_j \end{cases}$	(9)
	$td(t_i, t_j) = \frac{dist(t_i, t_j)}{dist_q(t_x, t_y)}$	(10)
	$TD = \sum_{k=1}^{m} dist_k$	(11)

The ranking measures produce three scores for each of the retrieved ontologies using following expressions described in Table 1. These scores must be weighted and combined to generate a single final ranking score for each of ontologies. The concept completeness measures the level of conceptualization of each matched concept. Generally, the level of conceptualization of a concept depends on the number of properties and relations with other concepts. This measurement assigns higher score to a concept having a relatively large number of properties. The score of concept completeness measurement is computed by the following expressions (5) and (6).

In expression (6), *tc(i)* denotes the concept completeness score of the concept *i*. *IC* and *EC* denote the number of internal properties and external properties respectively. The relation complexity measures how well concepts are interconnected based on sematic relations, such as superclass, subclass, association, and so on. This measurement, like the concept completeness measurement, also represents the level of conceptualization of ontology. We identify five types of relations. They are superclass, subclass, instance-of, sibling and association, which exist between concepts in ontology. We compute the concept relation complexity of the matched concepts using the following two expressions.

Lastly, the concept density measures how many intermediate concepts between matched concepts are existed. We define concept density measurement to find the best matching candidate from the list of ontologies. The best matching candidate denotes an ontology, which can extend the semantic structure and enhance semantic quality of query ontology after matching and merging. Thus, matched concept pair having longer distance has higher concept density score, which means better conceptualization. Following expression (9), (10) and (11) are evaluated to produce the score.

Let $t_i \rightarrow t_j$ be a shortest path between concepts t_i and t_j. Distance between the concepts can be computed using the expression (9). If query ontology has two concepts t_x and t_y that they are matched t_i and t_j to each other, the concept density score of a pair of concepts t_i and t_j can be computed by evaluating following expression (10). In this expression, $dist_q(t_x, t_y)$ denotes distance between the concepts t_x and t_y in query ontology. Let $dist_k$ be the concept density score of a pair of two concepts in a retrieved ontology. Following expression (11) produce the accumulated score of each concept density score for a program content ontology.

These three ranking scores should be normalized, and combined in order to make a single representative score for each of the ontologies in the ranking list. We normalize *TC*, *TRC*, and *TD* by virtue of ratio estimation as follows:

$$TC_{normal} = TC / TC_{max} \tag{12}$$

$$TRC_{normal} = TRC / TRC_{max} \tag{13}$$

$$TD_{normal} = TD / TD_{max} \tag{14}$$

4. Experiment

As showed in Table 2, we prepared program ontologies and usage histories of 53 TV program contents which are documentary programs dealing with various subjects, such as nature, science, animal, and culture to evaluate the performance of our approach. We set up usage histories, of which an item is formed with a tuple <uid, bdate, title, subtitle, prog_stime, prog_etime, watch_stime, watch_etime, genre, subgenre>, for 3 viewers.

Table 2. A Partial List of Experiment Data

Group	No.	Program Title	Subject
A	1	Kingdom of Animal	Animal
	4	Wilde Beast of Africa	Animal
	8	Chimpanzees of Tanganyika	Animal
B	9	Wildebeest Migration	Animal
	15	Queen of Africa	Culture
C	17	Insight Asia: Noodle Road	Culture
	35	Into Science	Science

We classified the collected documentary TV programs into three groups according to their similarity in terms of contents. Group A has documentary programs which describe similar subjects and contain same a few concept terms. Group B has documentary programs which represent different subjects but same a few concept terms. Group C has documentary programs which contain different subjects and. We used episodes of “kingdom of animal” series and “people of Kyrgyzstan” as query ontologies. We measured the performance of searching and ranking using precision, recall, and f-measure measurements.

Figure 6 represents the result of program searching experiments. The left grape shows the precision values of five queries for each viewer. The right graph shows precision, recall, and f-measure values of our proposed searching approach and keyword-based searching approach for two queries.

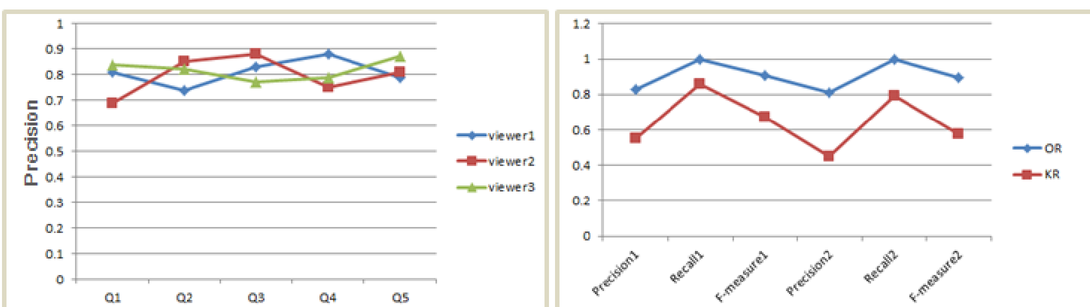

Figure 6. (left) Precision Values of Five Queries for Each Viewer (right)Comparison between Our Approach(OR) and Keyword-based Approach(KR) for Two Queries

From experimental result and performance evaluation, we found that our approach retrieved all relevant programs for given query data ontologies in spite of 80% precision rate. The cause of lower precision than recall is that some documentary programs irrelevant to query data are included in the search result because they have similar keywords to query data. Keyword-based approach has lower precision that our approach because irrelevant programs, which have same keywords but different subjects to query data, are included in the search result. This means that a few keywords only cannot represent core concepts of the contents of broadcasting programs.

5. Conclusion

In this paper, we present a new scheme for searching TV programs based on TV program ontologies and usage history examination. Our experiments for searching documentary programs prove that ontology-based searching is more precise than keyword-based searching in comparison of contents semantically. To speed up ontology-based broadcasting program search, the precise statistical data of each of ontologies must be stored and used. Thus our future work is the development of automatic searching process and reuse of statistical data.

Acknowledgements

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (No. 2011-0014613).

References

[1] ATIS IPTV Exploratory Group Report and Recommendation to the TOPS Council, http://www.atis.org/tops/IEG/ATIS_IPTV_EG_RPT_final.pdf.
[2] J. Xu, L. Zhang, H. Lu and Y. Li, "The development and prospect of personalized TV program", In Proceedings of the IEEE fourth International Symposium on Multimedia Software Engineering, **(2002)**, California, USA.
[3] P. Cotter and B. Smyth, "PTV: intelligent personalized TV guides", In Proceedings of the 12th Innovative Applications of Artificial Intelligence Conference, **(2000)**, Austin, Texas, USA.
[4] H. Zhang and S. Zheng, "Personalized TV program recommendation based on TV-anytime metadata", In Proceedings of the Ninth International Symposium on Consumer Electronics, **(2005)**, pp. 242-246.
[5] F. S. da Silva, L. G. P. Alves and G. Bressan, "PersonalTVware: An Infrastructure to Support the Context-Aware Recommendation for Personalized Digital TV", Int. Journal of Computer Theory and Engineering, vol. 4, no. 2, **(2012)**, pp. 131-135.
[6] E. Kim, S. Pyo, E. Park and M. Kim, "An Automatic Recommendation Scheme of TV Program Contents for (IP)TV Personalization", IEEE Tran. on Broadcasting, vol. 57, no. 3, **(2011)**, pp. 674-684.
[7] Y. Zhang, W. Vasconcelos and D. Sleeman, "Ontosearch: An ontology search engine", In Proceedings of the Int. Conf. on Innovative Techniques and Applications of Artifial Intelligence, **(2004)**, Cambridge, UK.
[8] P. Buitelaar, T. Eigner and T. Declerck, "OntoSelect: A Dynamic Ontology Library with Support for Ontology Selection", In Proceedings of the International SemanticWeb Conference, **(2004)**, Hiroshima, Japan.
[9] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi and J. Sachs, "Swoogle: A semantic web search and metadata engine", In Proceedings of the 13th ACM Conf. on Information and Knowledge Management, **(2003)**.
[10] C. Petel, K. Supekar, Y. Lee and E. K. Park, "OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification", In Proceedings of the Workshop On Web Information And Data Management, **(2003)**, pp. 58-61.

Authors

Jungmin Kim

Jungmin Kim is a professor in the Department of Computer Engineering at Daejin University, Pocheon, Korea. He received a B.S. degree and a M.S. degree in Computer Science in Hongik University in 1992 and 1994 respectively. He received a Ph.D. degree in Computer Engineering from Seoul National University in 2007. His research interests include Semantic Web, Semantic Information Processing, IPTV, and Smart Learning.

Hyunsook Chung

Hyunsook Chung is a professor in the Department of Computer Engineering at Chosun University, Gwang-ju, Korea. She received a B.S. degree in Physics in Daegu Catholic University and a M.S. degree in Computer Engineering from same university. She received a Ph.D. degree in Computer Engineering from Yonsei University in 2003. Her research interests include multimedia data processing, knowledge data processing, ontology engineering, and information processing.

Gukboh Kim

Dr. Guk-Boh Kim is currently a professor in the Department of Computer Engineering, DaeJin University at Pocheon in Korea. He received the M.S degrees in Computer science from Yensei University in Seoul, Korea and the Ph.D degree in Computer Science from Catholic University of Daegu in Korea 1998 respectively. He spent two years as a Director of Central Computer Center in Korean Navy and also spent three years as an assistant professor of Pukyong National University in Korea before joining the Daejin University. He was visiting professor at the University of Missouri at Rolla from 1999 to 2000. He was Head of computer Center in Daejin University. His research interests are in the following areas: software engineering, system analysis and design, and e-Biz software system.

* Corresponding Authors : Hyunsook Chung, Gukboh Kim