A Collaborative Relevance Feedback Approach to Task-driven Recommendation

I-Chin Wu Duen-Ren Liu

Institute of Information Management
National Chiao Tung University
Hsinchu, Taiwan
nancy.iim89g@nctu.edu.tw dliu@iim.nctu.edu.tw

Abstract

Managing knowledge is an important issue for organizations to gain sustainable advantage in today's competitive environments. Nevertheless, the operations of enterprises are mainly planned around task execution. Accordingly, it is important to provide knowledge support for task execution by recommending task relevant knowledge to fit the information needs of task execution. This paper contributes to the stream of knowledge management system by proposing a novel task-driven recommender system to take the merits of information retrieval and cooperative work. A task-oriented information repository based on fuzzy classification is deployed to support task-driven recommendation. Additionally, the system uses task specification to model the key contents of executing-task and to further facilitate the process of identifying relevant information needs of task execution. A collaborative relevance feedback approach is proposed to adjust task specification with the aid of cooperative workers and domain experts. The proposed system enhanced with the adaptation capability to adjust task specification makes the recommendation of relevant information more flexible and adaptable to dynamically-changing working environments.

Keywords

Knowledge management, task specification, relevance feedback, fuzzy linguistic approach, recommendation

1. Introduction

Managing Knowledge within and across organizations is considered as a prominent activity for creating sustainable competitive advantages in today's business environments. Knowledge Management (KM) is a cycle, sometimes repeated process, which generally includes creation, management and sharing activities. (Davenport & Prusak 1998, Fischer & Ostwald 2001, Gray 2001, Nonaka 1994, Wiig 1993). Information technology (IT) supports different types of knowledge management systems by adopting tools, including Data Warehouse, OLAP, Data Mining, Document Management System, Expert Systems and so on (Hahn & Subramani 2000). Among these ITs, document management plays an important role, since textual data such as articles, reports, manual, know-how documents and the like are treated as valuable and explicit knowledge within organizations (Nonaka 1994). To support effective knowledge (document) management, knowledge (information) retrieval is a core component (Gartner Group 1999) of KMS in providing knowledge guides or recommendations to people.

Generally, the operations of enterprises are mainly planned around tasks. For complex tasks, knowledge workers may need to work out problems collaboratively. As knowledge is embedded in task execution, providing task relevant knowledge (documents) to fit the information needs of task execution is important to support effective knowledge management. Recently, information retrieval (IR) technique has been considered in workflow management systems to assist knowledge workers find task relevant knowledge (Abecker et al. 2000, Fenstermacher 2002). However, these works did not consider providing task relevant knowledge collaboratively.

For complex and knowledge-intensive tasks, effective knowledge support requires the collaboration among knowledge workers. Sharing knowledge with peer groups is important in knowledge management (Fischer & Ostwals 2001). The Computer-Supported Cooperative Work (CSCW) (Rodden 1991) and recommender systems (Resnick & Varian 1997) also shade additional light on collaboration. CSCW emphasizes on the power of computer system to help group of people perform the tasks in a shared environment (Ellis 1991, Rodden 1991). Recommnder Systems employ content-based filtering and collaborative filtering to recommend web pages, movies, books and so on (Goldberg et al. 1992, Konstan & Riedl 2000, Pazzani 1999, Resnick et al. 1994, Schafer et al. 2000).

This work extends the idea of CSCW and recommender systems to provide effective knowledge support in collaborative and interactive working environments. A collaborative task-driven recommender system is proposed to provide knowledge (document) recommendation to support workers in the execution of tasks. A task-oriented information repository based on fuzzy classification is deployed to support the operation of recommendation. Moreover, to facilitate task-driven recommendation, the task specification of executing-task is the kernel of the recommender system to discover relevant information for recommendation. Notably, task specification describes the key content of executing-task that the worker conducts at hand. We propose a collaborative relevance feedback approach to adjust the task specification by the aid of cooperative workers and domain experts with a fuzzy linguistic approach. The recommendation is finally carried out to gathering information needs for executing-task based on the task specification.

The remainder of this paper is organized as follows. Section 2 presents the architecture of the proposed collaborative task-driven recommender system. The process of building task-oriented information repository by content analysis is presented in Section 3. Section 4 illustrates the proposed collaborative relevance feedback approach to generate and modify task specification. The capability of our proposed system to provide task-driven recommendation is demonstrated in Section 5. Conclusions and future work are finally discussed in Section 6.

2. Generic View of System Architectures

The aim of this work is to model, design and implement a collaborative task-driven recommder system that can determine task relevant knowledge from the view of tasks and user information needs. We broadly refer a task as a project, research work, or activity. Figure 1 shows the overview of the proposed system. Participators include knowledge workers engaged in a specific task and domain experts of a specific subject. Three main

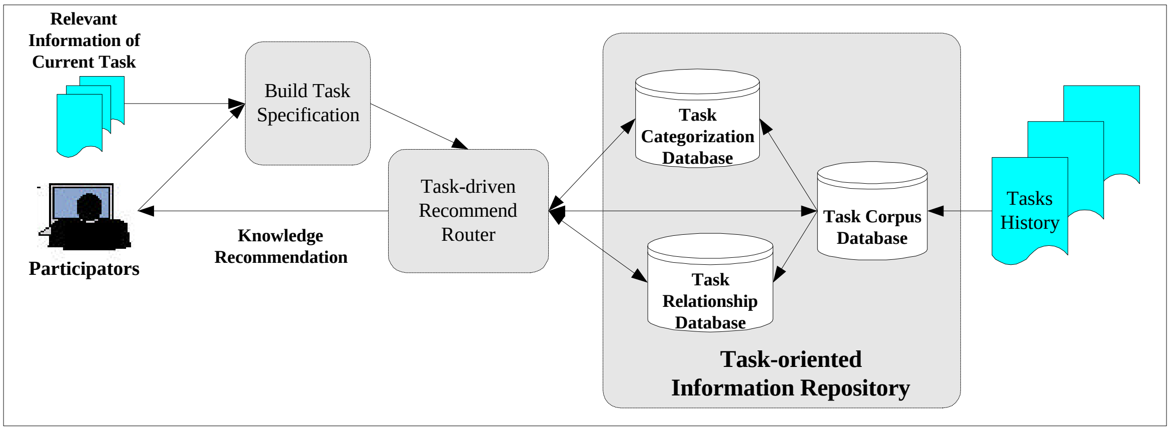
Relevant Information of Current Task
Build Task Specification
Task-driven Recommend Router
Task Categorization Database
Task Corpus Database
Task Relationship Database
Tasks History
Participators
Knowledge Recommendation
Task-oriented Information Repository

Figure 1. Overview of Collaborative Task-driven Recommender System

modules of the system, including task-oriented information repository, task specification and task-driven recommender router, are illustrated as follows.

- ***Task-oriented information repository.*** The information repository stores the information corresponding to task execution, and contains three databases including *task corpus*, *fuzzy task categorization*, and *fuzzy task similarity database*. The *task corpus* database stores the key profile of each task. A task corpus of a task t_r is represented as a set of keywords derived by analysing the set of documents that are generated and accessed by t_r. The *fuzzy task categorization database* records the fuzzy relationship of tasks and categories. Notably, tasks are categorized by fuzzy classification; thus, a task may belong to more than one category. The *fuzzy task similarity database* stores the similarity measures between tasks that are modelled as a fuzzy task similarity matrix. The repository is the basis for task-driven recommendations.
- ***Task specification.*** Task specification describes the key features of task conducted at hand and is the kernel to route task-relevant information to knowledge workers. The system generates task specification based on task corpus and employs a collaborative relevance feedback approach to adjust task specification.
- ***Task-driven recommend router.*** Task-driven recommender router assists knowledge workers gather proper information from the information repository. The router determines the task-relevant information according to the task specification.

The *task-oriented information repository* provides sufficient task relevant information for the system. The contents of textual data such as articles, reports, manual, know-how documents and so on are analyzed and stored in the information repository. Section 3 illustrates the processing of task relevant documents and the extraction of task corpus. *Task specification* describes the key contents of the executing task that the worker conducts at hand. The procedure of building task specification takes the merits of content-based and collaborative approach that lead to adaptable task specification. Task specification is constructed based on the task corpus. Moreover, a collaborative relevance feedback approach is proposed to adjust the task specification, as described in the following. A two-phase assessment approach is employed to select a set of referring tasks, and then to further evaluate the relevance between the executing task and referring tasks. The approach is conducted by the aid of knowledge workers and domain experts with a fuzzy linguistic approach. Based on the relevance evaluation, the task specification is adjusted by considering the relevance of referring tasks and their corresponding task corpus. More details are discussed in Section 4.

The *task-driven recommender router* plays the role of matching task specification with information repositories to streamline the process of information retrieval. The router retrieves and fetches relevant information based on the task specification. The process of recommendation will be discussed in Section 5. For the rest of this work, an auxiliary example of research tasks in a research department is presented to describe our proposed approach.

3. Constructing Task-Oriented Information Repository

This section addresses three essential phases to construct a task-oriented information repository: extracting task corpus from textual data collected during task execution; categorizing tasks into the information repository by a fuzzy classification approach; and calculating the fuzzy similarity among tasks.

3.1 Extracting Task Corpus

There are numerous tasks carried out in the organization and a huge amount of documents are generated and retrieved during task execution. Documents are stored according to which tasks they are generated from. Task corpora are generated by extracting knowledge embodied in textual documents.

Pre-Processing. Each document is represented in a *n*-dimensional vector space model, called document feature vector. The *term transforming* and *term weighting* steps are employed to find the most discriminating words among a set of documents (Baeza-Yates & Ribeiro-Neto 1999). It is impossible to use every term that appear in the document to index a document. The *term transforming* step includes case folding, stemming, and stop word removing. The *tf-idf* is a well-known approach for term (keyword) weighting in the information retrieval literature (Salton 1988). The weight of a keyword *i* in a document *d* is defined as term frequency $tf_{i,d}$ multiplied by the inverse document frequency, idf_i, i.e., log N/df_i, as shown in the following equation.

$$w_{i,d} = tf_{i,d} \times \log \frac{N}{df_i} \tag{1}$$

where idf_i is inverse document frequency, N is the number of documents within a task, and df_i is the number of documents that contain the ith term. This leads to the set of discriminating words to represent documents, $KY=\{kw_1, kw_2, \cdots, kw_n\}$. The document feature vector of document d is represented as, $\vec{\mathbf{d}} =< w_{1,d}, w_{2,d}, \cdots, w_{n,d} >$.

Task Corpus Generated. We define the **task corpus** of task t_r as the **centroid** vector $\vec{\mathbf{p}}_{t_r}$ which is the vector obtained by averaging the weights of keywords in the set of documents that are generated and accessed during the execution of task t_r. A centroid of a set of documents D_{t_r} is given by Eq. (2). The task corpus database stores a set of task corpus.

$$\vec{\mathbf{p}}_{t_r} = \frac{1}{\left|D_{t_r}\right|} \sum_{d \in D_{t_r}} \vec{\mathbf{d}} \tag{2}$$

3.2 Fuzzy Task Categorization

Fuzzy classification extends traditional crisp classification notation to associate each objects in every category with a membership function so that each task can belong to more than one category (Zadeh 1965). In our proposed method, the relevant degrees between tasks and categories are calculated based on the keyword distribution in each category.

Domain experts pre-define a set of categories by manually selecting categories that represent subjects in an organization. Furthermore, the seed-based technique is applied to generate the concept of category, as described in the following. Experts select seed tasks from a set of task corpus, $TP = \{\overset{r}{\mathbf{p}}_{t_1}, \overset{r}{\mathbf{p}}_{t_2}, \mathrm{L}\ , \overset{r}{\mathbf{p}}_{t_k}\}$ to represent the concept of a category. Notably, the selected tasks stand for the category they belong to, and thereby, are called the seed tasks of the category.

Modeling Keyword-Category Space. Keyword distributions in categories can be derived from the seed tasks represented in feature vectors, i.e., task corpora. Definition I defines the category descriptor probability matrix ***C*** representing the probabilities that keywords appear in categories.

<u>*Definition I*:</u> Let *X* be a set of categories, *X*={c_1, c_2,L , c_m}, and *KY* be a set of most descriptive keywords, *KY*={kw_1, kw_2,L , kw_n}. *TP* be the set of seed tasks to denote the categories. $=[\ _{mn}]$ denotes a *m-by-n* matrix, in which an element $_{ij}$ ($_{ij} \in [0,1]$) in the matrix is the probability distribution that the *j*th keyword appears in the *i*th category, which is computed from keyword distribution in *TP*.

Modeling Task-Category Space. According to the category descriptor probability matrix , we can determine the membership grades of *r*th task in the *i*th category, $m_{c_i}(t_r)$, as defined in Eq. (3),

$$m_{c_i}(t_r) = \frac{\sum_{j=1}^{n} h_{rj}\hat{c}_{ij}}{\sum_{j=1}^{n} h_{rj}} \qquad h_{rj} = \begin{cases} 1 & t_r \ \textit{contain keyword}\ k_j \\ 0 & \textit{otherwise} \end{cases} \tag{3}$$

where $m_{c_i}(t_r) \in [0,1]$; *j* denotes the *j*th keyword; *n* is the number of discriminating keywords; and h_{rj} is a binary representation.

A fuzzy relation matrix ***R***, as defined in Definition II, represents the relevance degrees between tasks and categories.

<u>*Definition II*:</u> Let *X* be a set of categories, *X*={c_1, c_2,L , c_m}, and let *T* be a set of tasks, *T*={t_1, t_2,L , t_k}. ***R***=[$m_{c_i}(t_r)$] denotes an *m-by-k* matrix, where an element $m_{c_i}(t_r)$ in the matrix denotes the relevance degree of *r*th task to the *i*th category, as derived from Eq. (3).

Task Representation. The fuzzy relation matrix ***R*** models the relevance degree between tasks and categories. A category c_i can be represented as a fuzzy subset *T* characterized by membership function:

$$\overset{\%}{c}_i = \sum\nolimits_{t_r \in T} m_{c_i}(t_r)/t_r, \ \ \text{T is a collection of tasks.} \tag{4}$$

where $m_{c_i}(t_r)$ is the relevance degree of t_r to c_i.

Example 1: Let $TP = \{\vec{\mathbf{p}}_{t_1}, \vec{\mathbf{p}}_{t_2}, \ldots, \vec{\mathbf{p}}_{t_{10}}\}$ denotes ten tasks corpus derived from ten tasks selected from a research department. $X=\{c_1, c_2, \ldots, c_4\}$ denotes four predefined categories in the repository. Notably, the description of tasks and categories are listed in Appendix I. The result calculated by Eq. (3) is shown as follows.

$$\boldsymbol{R} = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{array} \begin{array}{c} \begin{array}{cccccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} \end{array} \\ \begin{bmatrix} 0.10 & 0.03 & 0.08 & 0.12 & 0.07 & 0.06 & 0.49 & 0.07 & 0.18 & 0.13 \\ 0.25 & 0.08 & 0.25 & 0.19 & 0.11 & 0.10 & 0.11 & 0.15 & 0.18 & 0.21 \\ 0.22 & 0.08 & 0.24 & 0.29 & 0.13 & 0.68 & 0.15 & 0.54 & 0.29 & 0.27 \\ 0.07 & 0.71 & 0.07 & 0.06 & 0.04 & 0.03 & 0.03 & 0.05 & 0.06 & 0.07 \end{bmatrix} \end{array}$$

From the above fuzzy sets, we can observe that some tasks are highly relevant to some categories, but others are not. Assume that a threshold $Thres_N > 0.10$ is used to filter out low relevant tasks in each category. According to the formula (4), we can express the category, c_i, as a fuzzy subset *T*, for example, $\tilde{c}_1 = 0.12/t_4 + 0.49/t_7 + 0.18/t_9 + 0.13/t_{10}$. The fuzzy task categorization database records the result of fuzzy classification. Each task is associated with a weight $m_{c_i}(t_r)$ representing the task relevance degree to category c_i.

3.3 Fuzzy Task Similarity

According to the task corpus expressed as a **centroid** vector $\vec{\mathbf{p}}$, we can calculate the similarity measure between tasks by cosine measure (Salton 1988) defined as follows:

$$sim(t_r, t_j) = \frac{\vec{\mathbf{p}}_{t_r} \cdot \vec{\mathbf{p}}_{t_j}}{\left|\vec{\mathbf{p}}_{t_r}\right|\left|\vec{\mathbf{p}}_{t_j}\right|} = \frac{1}{\left|\vec{\mathbf{p}}_{t_r}\right|\left|\vec{\mathbf{p}}_{t_j}\right|}\sum_{i=1}^{n} w_{i,t_r} \times w_{i,t_j} \qquad (5)$$

$\left|\vec{\mathbf{p}}_{t_r}\right|$ and $\left|\vec{\mathbf{p}}_{t_j}\right|$ are Euclidean length of $\vec{\mathbf{p}}_{t_r}$ and $\vec{\mathbf{p}}_{t_j}$, respectively. w_{i,t_r} and w_{i,t_j} are the weight of *i*-th term in t_r and t_j, respectively. The fuzzy task similarity matrix can be defined as in definition III.

<u>*Definition III*:</u> A fuzzy task similarity matrix records similarity measures between tasks, in which an element $sim(t_r, t_j) \in [0,1]$ in represents the similarity measure between t_r and t_j. is a reflexive and symmetric matrix.

4. Deriving Task Specification

This section illustrates the process of building task specifications based on task corpora and users perspectives, as shown in Fig. 2. An executing task denotes either a new task or an existing task that the knowledge worker conducts at hand. The task specification of the executing-task is initially derived by analysing the key contents of the task or by retrieving from the corresponding task corpus. Moreover, a collaborative relevance feedback approach is employed to adjust the task specification. The approach first uses a two-phase assessment to identify the information needs of the executing-task, and then uses a relevance feedback technique to adjust the task specification.

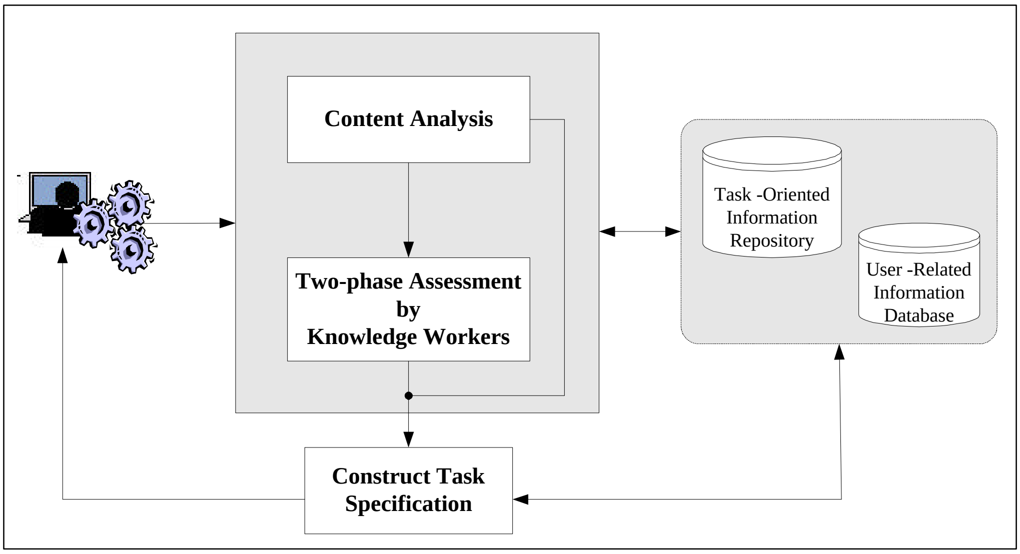
Content Analysis
Two-phase Assessment by Knowledge Workers
Construct Task Specification
Task -Oriented Information Repository
User -Related Information Database

Figure 2. The Process of Building Task Specification.

For the two-phase assessment, evaluators including knowledge workers and domain experts collaborate to assess the relevance between the executing-task and referring tasks by a fuzzy linguistic approach. Based on the relevance evaluation, the task specification is adjusted by a relevance feedback technique which is modified from the standard Rocchio approach. The result can be feedback to the system to adjust the task-corpus database. In addition, relevance degrees among tasks and categories will be modified.

4.1 Two-Phase Assessment by a Fuzzy Linguistic Approach

Knowledge workers may have difficulty to exactly specify their information needs, but they can identify information needs by referring to relevant tasks. In the following, a two-phase assessment approach is presented to systematically model the procedure of relevance assessment via the collaboration of cooperative workers. The fuzzy linguistic approach is applied to assess the relevance degree of tasks and categories in a more humanity way.

4.1.1 Phase1: Assessing Relevant Categories

As described by Zadeh (1975), it is very difficult to precisely assess information with a quantitative form, so using the notion of a linguistic variable is helpful. Fuzzy linguistic approach is an approximate technique to model human thinking and help human decision-making. In the following, we apply fuzzy linguistic approach to obtain the common consensus among cooperative workers and experts.

DefinitionIV (Jang et al. 1997, Zadeh 1975): A **linguistic variable** is characterized by a quintuple (*S*, *E(S)*, *U*, *G*, *M*) in which *S* is the *name* of the variable; *E(S)* is the *linguistic terms* of *S*, that is the set of its linguistic values range over universe of discourse *U*. *G* is a *syntactic rule* (a grammar) which generates linguistic term set in *E(S)*; and *M* is a *semantic rule* which assigns meaning, *m(ø)*, to each linguistic term *ø* in *E* with a fuzzy set on *U*.

Step1: Determine the semantic term set and corresponding fuzzy number. According to the definition IV, a linguistic variable, *Relevance*, is defined to represent the relevant degree between items (tasks or categories) that are assessed by evaluators. *E(Relevance)* is characterized by a fuzzy set of a universe of discourse *U*=[0,1], in which six linguistic terms and their associative semantic meanings are defined as follows.

E(Relevance) = { $ø_0$ = *very low*, $ø_1$ = *low*, $ø_2$ = *normal*, $ø_3$ = *high*, $ø_4$ = *very high*, $ø_5$ = *perfect*}
where $m(ø_i) < m(ø_j)$ for $i < j$, and all $m(ø_j)$ are distributed in [0,1].

The anti-symmetric distributed term set (Herrera-Viedma 2001) is adopted in our approach, where more positive linguistic terms are defined, as shown in the defined term set, since we put more emphasis on positive feedback to items. The anti-symmetric distributed term set is defined such that a sub domain may be more important and informative than the rest sub domain. In addition, we employ TFN to express the fuzzy scale of each linguistic term (see Figure 3). The triangular fuzzy number (TFN), defined in Definition V, is wildly used owing to their simplicity and solid theoretical basis (Pedrycz 1994), and thus is used to represent each linguistic term of the "Relevance" variable.

<u>*Definition V*</u> (Dubis& Prade 1978): A fuzzy number $\tilde{Z}$ is a "normal" and "convex" fuzzy subset defined on the set ¡ and its membership function is $f_{\tilde{Z}}(x) \in [0,1]$. The membership function $f_{\tilde{Z}}(x)$ of the triangular fuzzy number (TFN) $\tilde{Z} = (l,\ m,\ r)$ is given by Eq.(6).

$$f_{\tilde{Z}}(x) = \begin{cases} (x-l)/(m-l) & l \le x \le m \\ (r-x)/(r-m) & m \le x \le r \\ 0 & otherwise \end{cases} \tag{6}$$

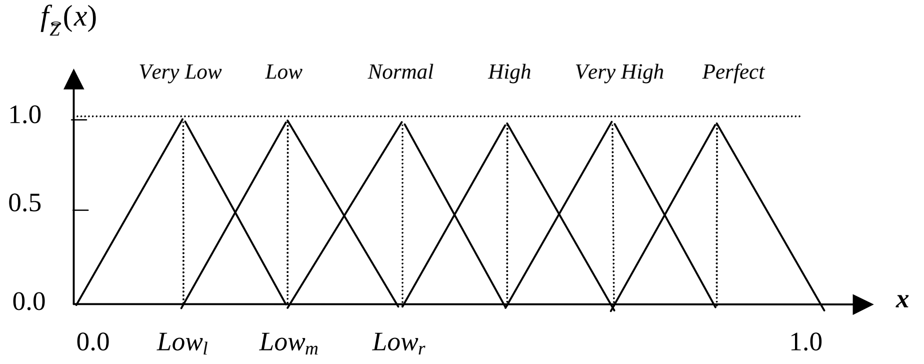

Figure3: Six Terms with Associative Semantic Meaning of "Relevance".

Step2: Assess the relevance to each category by evaluators. In this step, evaluators assess the relevance of the executing task to each category by linguistic ratings from their perceptions. Notably, evaluators may not have identical fuzzy numbers on six linguistic terms of "Relevance" due to different perceptions on the linguistic terms, as shown in Table 1. The assessment of the relevance to categories will be illustrated in Example 2.

Step3: Defuzzification. The fuzzy linguistic approach models the meaning of each term by fuzzy numbers. For computational advantage, we need to extract crisp rating (Best Non-fuzzy Performance values; BNP) from fuzzy numbers. There is a variety of methods to defuzzify fuzzy numbers such as mean of maximal (MOM), center of area (COA), Bisector of area (BOA), Small of maximum (SOM), Largest of maximum (LOM), and so on. (Jang et al. 1997). We adopt the COA method to calculate fuzzy numbers since the COA method is simple and practical and it calculates the fuzzy mean under uniform probability distribution assumption (Lee & Li, 1988). In particular, if the fuzzy number $\tilde{Z}$ is triangular, it is easy to obtain the crisp value (rating) by the Eq. (7).

$$CV(\tilde{Z}) \ = [(r-l)+(m-l)]/3+l \tag{7}$$

Table 1: Corresponding Fuzzy Numbers of Linguistic Term Set.

	VL	L	N	H	VH	P
Evaluator$_1$	(0,0.2,0.4)	(0.3,0.4,0.5)	(0.4,0.5,0.6)	(0.5,0.6,0.7)	(0.6,0.7,0.8)	(0.7,0.8,0.9)
Evaluator$_2$	(0,0.1,0.2)	(0.1,0.3,0.5)	(0.4,0.5,0.6)	(0.5,0.6,0.7)	(0.6,075,0.9)	(0.7,0.9,1)
Evaluator$_3$	(0,0,0)	(0.1,0.25,0.4)	(0.3,0.4,0.5)	(0.6,0.7,0.8)	(0.7,0.8,0.9)	(0.8,0.9,1)
Evaluator$_4$	(0,0,0)	(0.1,0.3,0.5)	(0.5,0.6,0.7)	(0.6,0.7,0.8)	(0.7,0.8,0.1)	(0.8,0.9,1)

Note: P=Perfect, VH=Very High, H=High, N=Normal, L=Low, VL=Very Low.

Table2: Evaluators Evaluate Relevance by Linguistic Terms.

	Evaluator$_1$	Evaluator$_2$	Evaluator$_3$	Evaluator$_4$
c$_1$	N	N	N	H
c$_2$	VH	H	N	VH
c$_3$	P	VH	VH	H
c$_4$	VL	N	L	L

Note: c$_i$ denotes the *i*th category.

Table 3: Crisp Values of Evaluators' Linguistic Ratings on Category

	Evaluator$_1$	Evaluator$_2$	Evaluator$_3$	Evaluator$_4$
c$_1$	0.5	0.5	0.4	0.7
c$_2$	0.7	0.6	0.4	0.8
c$_3$	0.8	0.75	0.8	0.7
c$_4$	0.2	0.5	0.25	0.3

Step4: Aggregate performance values of evaluators. The ordered weighted average operator (Yager 1988) is employed to integrate the crisp ratings (BNP values) of evaluators' linguistic ratings. Let $A_{ej}(c_i)$ be the evaluator e_j's crisp rating on the relevance of the executing task to category c_i. Let w_{ej} be the associated weight representing the relative importance (weight) of evaluator e_j's rating. The aggregated relevance of the executing task to category c_i, $A_E(c_i)$, is derived as $\sum_j w_{ej}A_{ej}(c_i)$. The evaluation result can be formulated as a fuzzy query vector $\vec{\mathbf{q}}_E$ which denotes the relevance degrees of the executing task to categories, where $\vec{\mathbf{q}}_E = < A_E(c_1), A_E(c_2), \cdots, A_E(c_m) >$

Example 2: Assume that a new task, "*Recommendation in Enterprises* ", is evaluated to derive the task specification. Four evaluators determine the relevance degrees of the new task to each category with linguistic ratings based on their subjective judgments, as listed in Table 2. Six linguistic scales of corresponding fuzzy numbers that are determined by the four evaluators are listed in Table 1. Table 3 shows the crisp values transformed from evaluators' linguistic ratings according to Eq. (7). If we set $w_{ej} = 1/n_e$, where n_e is the number of evaluators, then the aggregated performance value is computed as the arithmetic mean:

$$A_E(c_1) = 0.25 \cdot 0.5 + 0.25 \cdot 0.5 + 0.25 \cdot 0.4 + 0.25 \cdot 0.7 = 0.525$$

$$A_E(c_2) = 0.25 \cdot 0.7 + 0.25 \cdot 0.6 + 0.25 \cdot 0.4 + 0.25 \cdot 0.8 = 0.625$$

$$A_E(c_3) = 0.25 \cdot 0.8 + 0.25 \cdot 0.75 + 0.25 \cdot 0.8 + 0.25 \cdot 0.7 = 0.7625$$

$$A_E(c_4) = 0.25 \cdot 0.2 + 0.25 \cdot 0.5 + 0.25 \cdot 0.25 + 0.25 \cdot 0.3 = 0.3125$$

The fuzzy query vector $\vec{\mathbf{q}}_E$ is expressed as follows.

$$\vec{\mathbf{q}}_E = < A_E(c_1), A_E(c_2), A_E(c_3), A_E(c_4) > = < 0.525, 0.625, 0.7625, 0.3125 >$$

Table 4: The Relevant Degree between Tasks and Categories.

	Category1	Category2	Category3	Category4	Similarity Measure	Ranking
Task 1	0	0.25	0.22	0	0.8388	(6)
Task 2	0	0	0	0.71	0.2694	(10)
Task 3	0	0.25	0.24	0	0.8440	(5)
Task 4	0.12	0.19	0.29	0	0.9468	(3)
Task 5	0	0.11	0.13	0	0.8499	(4)
Task 6	0	0	0.68	0	0.6574	(9)
Task 7	0.49	0.11	0.15	0	0.7244	(8)
Task 8	0	0.15	0.54	0	0.7776	(7)
Task 9	0.18	0.18	0.29	0	0.9566	(1)
Task 10	0.13	0.21	0.27	0	0.9551	(2)

Note: **Similarity Measure** denotes the similarity degree between executing task and existing task.

4.1.2 Phase 2 Selecting the Referring Tasks

Step1 Select relevant categories and calculate similarity values. In *Example 2*, a query vector $\vec{\mathbf{q}}_E$ is derived, $\vec{\mathbf{q}}_E =< 0.525, 0.625, 0.7625, 0.3125 >$. The similarity measure between the executing (new) task t_{exe} and existing task t_r can be computed according to $\vec{\mathbf{q}}_E$ and the fuzzy relation matrix ***R***, as shown in Eq. (8), where $\vec{\mathrm{t}}_{t_r} =< m_{c_1}(t_r), m_{c_2}(t_r), \ldots, m_{c_m}(t_r) >$. Notably, ***R*** records the relevance degrees between tasks and categories. By ranking the similarity measures, the top-N/last-N tasks are selected as the positive/negative referring tasks.

$$sim(t_{exe}, t_r) = \frac{1}{|\vec{\mathbf{q}}_E||\vec{\mathrm{t}}_{t_r}|}\sum_{i=1}^{m} A_{\mathrm{E}}(c_i) \times m_{c_i}(t_r) \qquad (8)$$

Example 3: The fuzzy query vector $\vec{\mathbf{q}}_E$ represents the relevance degrees between the new task and categories. The result of membership degree of *r*th task in the *i*th category is listed in Table 4. The similarly measures of new task and existing tasks derived by Eq. (8) are shown in the fourth column of Table 4. The ranking of similarity measures is shown in the fifth column of Table 4. The top-3 tasks, t_4, t_9 and t_{10}, are selected as the positive referring tasks, while the last-3 tasks, t_2, t_6 and t_7, are selected as the negative referring tasks. The referring tasks will be used for further recommendation in Section 5.

Step2 Evaluate the relevance of referring tasks by fuzzy linguistic approach. The evaluators assess the relevance degree between the executing task and referring tasks without reviewing all tasks. Table 5 shows the result of the assessment. Notably, the procedure of task assessment is similar to the procedure of category assessment. The ordered weighted average operator is employed to integrate the crisp ratings (BNP values) of evaluators' linguistic ratings. Let $A_{ej}(t_r)$ be the evaluator e_j's crisp rating on the relevance of the executing-task to task t_r. Let w_{ej} be the associated weight representing the relative importance (weight) of evaluator e_j's rating. The aggregated relevance of the executing-task to task t_r, $A_E(t_r)$, is derived as $\sum_j w_{ej}A_{ej}(t_r)$.

Table 5: Tasks are Evaluated by Linguistic Variables.

	Expert$_1$	**Expert$_2$**	**Expert$_3$**	**Expert$_4$**	**Keyworker$_1$**	**Keyworker$_2$**
Candidate task set has positive relationship with a new task						
Task 4	P	H	H	L	P	P
Task 9	VH	VH	VH	H	H	P
Task 10	N	N	N	H	H	N
Candidate task set has negative relationship with a new task						
Task 2	VL	VL	VL	L	Z	L
Task 6	L	L	VL	L	Z	N
Task 7	VL	L	L	Z	L	VL

4.2 Deriving Task Specification by Relevance Feedback

In general, users may not be able to specify their queries precisely. Thus, various relevance feedback techniques have been proposed to formulate and modify a query by assessing the relevance of documents. Rocchio (1971) is a classic relevance feedback method to formulate a query expressed in vector space model, as shown in Eq. (9).

$$\text{Standard\_Rocchio: } \vec{\mathbf{q}}_m = a\,\vec{\mathbf{q}} + b\,\frac{1}{|D_r|}\sum_{\forall \vec{\mathbf{d}}_j \in D_r} \vec{\mathbf{d}}_j - g\,\frac{1}{|D_n|}\sum_{\forall \vec{\mathbf{d}}_j \in D_n} \vec{\mathbf{d}}_j \qquad (9)$$

D_r is the set of relevant documents and D_n is the set of irrelevant documents, which are accessed according to the users' judgement. $|D_r|$ and $|D_n|$ stand for the number of documents in the sets D_r and D_n, respectively. a, b, and g are tuning constants, Rocchio (1971) set $a = 1$ and Ide(1971) set $a = b = g = 1$.

Knowledge workers may not be able to precisely express their information needs (represented as task specification) in the initial stage of task execution. Thus, we adopt the relevance feedback technique to adjust task specifications. In this work, we modify the standard Rocchio approach by adding the referring tasks' relevance degrees that are obtained from the fuzzy linguistic assessment. The modification considers the relative importance of relevant (positive) and irrelevant (negative) tasks from users' viewpoints. Two aspects are considered, one is constructing task specifications for new tasks and the other is adjusting task specifications for existing tasks. Notably, the proposed system finds task-relevant information based on the task specification..

4.2.1 Constructing Task Specification for New Task

Eq. (10) shows the proposed formulation, modified from the standard Rocchio formulation, to construct a task specification of new task. The formulation mainly constructs the task specification based on the task corpuses of referring tasks with associated weights derived from the task assessment.

$$\vec{\mathbf{s}}_{new} = a\,\vec{\mathbf{s}}_{initial} + b\sum_{\forall t_j \in T_r}(w_{t_j})\vec{\mathbf{p}}_{tj} + g\sum_{\forall t_j \in T_n}(1-w_{t_j})\vec{\mathbf{p}}_{tj} \qquad (10)$$

$\overset{1}{\mathbf{s}}_{initial}$ is the initial specification derived from analysing collected relevant documents for the new task, if available. $\overset{1}{\mathbf{s}}_{initial}$ is an optional part in the equation. T_r is the set of relevant (positive) tasks and T_n is the set of irrelevant (negative) tasks according to the assessment of experts and workers, as described in section 4.1. $\overset{1}{\mathbf{p}}_{tj}$ is the task corpus of task t_j. w_{tj} is the associated weight of task t_j representing the relevance degree of the new task to t_j. w_{tj} is set to $A_E(t_j)$, the aggregated relevance of the new-task to task t_j, which is derived from the task assessment procedure illustrated in Section 4.1.2. Finally, a , b and g are tuning constants.

4.2.2 Adjusting Task Specification

The section illustrates the adjustment of task specifications for existing tasks based on the relevance feedback approach. Existing tasks can also be evaluated to derive the relevance degrees to referring tasks according to the two-phase linguistic assessment. Notably, the similarity measures between existing tasks have already been computed and stored in the fuzzy task similarity matrix . The assessment can simply start from phase 2 that uses the task similarity measures recorded in to select the set of referring tasks.

Eq. (11) shows the proposed formulation, modified from the standard Rocchio formulation, to adjust the task specification of existing task. The formulation revises the task specification based on the task corpuses of referring tasks.

$$\overset{1}{\mathbf{s}}_{rev} = a\overset{1}{\mathbf{p}}_{t_{rev}} + b\sum_{\forall t_j \in T_r} (o_{t_j})\overset{1}{\mathbf{p}}_{tj} + g\sum_{\forall t_j \in T_n} (1 - o_{t_j})\overset{1}{\mathbf{p}}_{tj}$$

$$where\ o_{t_j} = l A_E(t_j) + (1 - l)sim(t_{rev}, t_j)\ ; \quad l \in [0,1] \qquad (11)$$

$\overset{1}{\mathbf{s}}_{rev}$ is the revised task specification of an existing task t_{rev}. $\overset{1}{\mathbf{p}}_{t_{rev}}$ is the original task corpus of t_{rev}. Different from Eq. (10), the associated weight, o_{tj}, of a referring task t_j combines both the similarity measure $sim(t_{rev}, t_j)$ recorded in and the aggregated relevance degree between task t_{rev} and t_j , $A_E(t_j)$, derived from the task assessment. λ is a parameter to tune the weight of $sim(t_{rev}, t_j)$ and $A_E(t_j)$.

Finally, the task specification of the executing task t_{exe} derived from equation (10) or (11) can be expressed as a set of discriminating terms, $\overset{1}{\mathbf{s}}_{t_{exe}} =< w_{1,exe}, w_{2,exe}, \mathrm{L}\ , w_{q,exe} >$. $w_{i,exe}$ stands for the associated weight of *i*th keywords to task t_{exe}. q is the total number of weighted discriminating terms of a task. The set of keywords are used further to retrieve relevant and proper information.

5. Recommendation

Based on the task specifications, the system recommends/retrieves relevant information from the task-oriented information repository to help knowledge workers execute tasks conducted at hand (executing tasks). The similarity measures between the executing-task and tasks stored in the information repository are computed to select top-N relevant tasks for recommendation. Moreover, knowledge workers engaged in relevant tasks, relevant documents and correlated keywords are recommended, as shown in Figure 4.

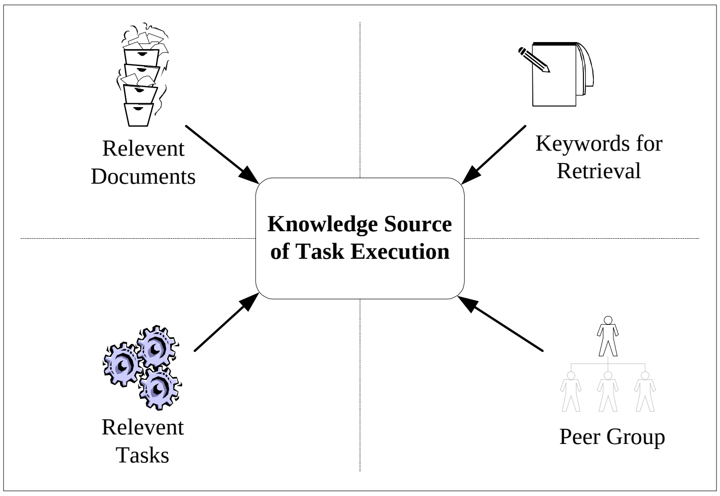

Figure 4: Recommended Relevant Items for the Executing Task

Table 6: Relevant Tasks and Peer Group of the Executing Task.

Task ID	Task Name	Key-workers ID	Domain Experts ID
Task 4	ERP Systems	Key-worker 3, Key-woker15	Expert 1,Expert 2
Task 8	Data Warehouse Deployment	Key-worker 4, Key-worker 6 Key-worker 7, Key-worker 8	Expert2, Expert4, Expert 5
Task 9	Web Logs Analysis	Key-worker 1, Key-worker 2,	Expert 1, Expert 3

Note: Task 8 is the newest found task that was not positive in the phase 2 of Section 4.1.2

Example 4: Let t_{exe} denote the executing task, i.e., the task (either a new task or an existing task) that the worker conducts at hand. For the new task, "*Recommendation in Enterprises*", the recommendation router provides the task-relevant information as follows.

Relevant Task and Peer Group Recommendation. As $\mathbf{s}_{t_{exe}}$ has been derived by Eq.(10)/Eq.(11), retrieving relevant tasks for references will be helpful. The cosine measure of $\mathbf{s}_{t_{exe}}$ and $\mathbf{s}_{t_j}$, i.e., *sim(*t_{exe}*,* t_j*),* is computed as the similarity measure between the executing task and task t_j. Those tasks with top-N similarity measures are selected as the relevant tasks for recommendation. Table 6 lists the relevant tasks and knowledge workers engaged in each relevant task

Document Recommendation. Let $\boldsymbol{M}_{\text{exe}} = [f_{1l}]$ denote a *one-by-l* task similarity matrix (i.e. similarity measures between the *executing task* and *l relevant tasks*); $\boldsymbol{N}=[g_{lk}]$ denote a *l-by-k* similarity matrix of *l* relevant tasks to *k* documents; and $\boldsymbol{RS}=[h_{1k}]$ denote the relevance degrees of the executing task to documents. $\boldsymbol{RS}=\boldsymbol{M}_{\text{exe}} \circ \boldsymbol{N}$. The matrix operation is defined in Eq. (12).

$$[h_{1k}] = [f_{1l}] \circ [g_{lk}],\ where\ h_{1k} = \max_{l} \min[f_{1l}, g_{lk}] \qquad (12)$$

In Eq. (12), the min and max operations, instead of the product and sum operations, are used in the matrix operation. The top-N relevant documents in ***RS*** are selected for recommendation.

Keyword Recommendation. The keyword set is derived from the constructed task specification $\vec{\mathbf{s}}_{t_{exe}}$. The system displays the keyword set to assist knowledge workers in conducting further retrieval. The keyword set forms the task corpus of the executing task, and may be modified by further executions of the task. Some of the seed corpuses and their associated weights are listed as follows. Notably, the keywords have been stemmed.

Recommendation in Enterprises**: (Web,** 0.42), (**mine**, 0.26), (**data**, 0.23), (**inform**, 0.19), (**busi**, 0.17), (**usag**, 0.17), **(member**, 0.16), (**transact**, 0.15), (**techniqu**, 0.15), (**person**, 0.11), (**role**, 0.11), (**site**, 0.11),L , (**design**, 0.10).

6. Conclusions and Future Work

In this work, the architecture of a task-driven recommender system is presented. Several perspectives are considered in the proposed system. The task-oriented information repositories are constructed based on the content analysis of textual data. The adaptable task specification is derived using a collaborative relevance feedback approach, in which cooperative workers and domain experts collaborate to determine the relevance of tasks. The proposed system enhanced with the adaptation capability to adjust task specification makes the recommendation of relevant information more flexible and adaptable to dynamically-changing working environments. Our work is currently towards implementing the proposed task-driven recommender system. In the future, experiments will be conducted to evaluate the effectiveness of the collaborative relevance feedback approach for adaptation of task specifications.

Acknowledgements

This research was supported by the National Science Council of the Republic of China under the grant NSC 91-2416-H-009-008.

References

Abecker, A, Bernardi, A, Hinkelmann, K, Kühn, O & Sintek, M (2000), 'Context-Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project' *Int. Journal on Information Systems Frontiers (ISF)*, 2(3/4), pp. 139-162, Kluwer.

Baeza-Yates, R & Ribeiro-Neto, B (1999), *Modern Information Retrieval,* Addison-Wesley, pp 15-25

Davenport, TH & Prusak, L (1998), *Working Knowledge: How Organizations Manages What They Know*, Harvard Business School Press.

Dubis, D, & Prade, H (1978), 'Operations on Fuzzy Numbers', *International Journal of Systems Science*, **9**(3), pp. 613-626.

Ellis, CA, Gibbs, SJ & Rein, GL (1991), 'Groupware: Some Issues and Experiences', *Communications of the AC*M, 34 (1), pp. 38-58.

Fenstermacher Kurt D (2002), 'Process-Aware Knowledge Retrieval', *Proc. of 35th Hawaii International Conference on System Sciences,* Big Island, Hawaii, USA, pp. 209-217.

Fischer, G & Ostwald, J (2001), 'Knowledge Management: Problems, Promises, Realities, and Challenges', *IEEE Intelligent Systems*, 16(1), pp. 60-73.

Gartner Group (Summer 1999), Knowledge Management Report.

Goldberg, D, Nichols, D, Oki, M, & Terry, D (1992), “Using Collaborative Filtering to Weave an Information Tapestry,” *Comm. of the ACM,* 35(12), pp.61-70.

Gray, PH (May 2001), ‘Problem-Solving Perspective on Knowledge Management Practices’, *Decision Support Systems*, 31(1), pp. 87-102.

Hahn, J & Subramani, M (2000), ‘A Framework of Knowledge Management Systems: Issues and Challenges for Theory and Practice’, *21st International Conference on Information Systems (ICIS)*, Brisbane, Australia, pp. 302-312.

Herrera-Viedma, E (April 2001), ‘Modeling the Retrieval of an Information Retrieval System Using an Ordinal Fuzzy Linguistic Approach’, *Journal of the American Society of Information Science (JASIST),* 52(6), pp. 460-475.

Ide, E (1971), ‘New Experiments in Relevance Feedback’, *In the SMART Retrieval System: Experiments in Automatic Document Processing (ed. Salton, G)*, pp. 337-354, Prentice Hall PTR.

Jang, JS R, Sun, CT & Mizutani, E. (1997), *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence***,** Prentice Hall PTR.

Lee, ES & Li, RL (1988), ‘Comparison of fuzzy number based on the probability measure of fuzzy events’, *Computer and Mathematics with Applications*, 15, pp. 887-896

Nonaka, I (1994), ‘A Dynamic Theory of Organizational Knowledge Creation’, *Organization Science*, 5(1), pp.14-37.

Pazzani, M (December 1999), ‘A Framework for Collaborative, Content-Based and Demographic Filtering’, *Artificial Intelligence Review,* 13(5-6), pp. 393-408.

Pedrycz, W (1994) ‘Why Triangular Membership Functions?’, *Fuzzy Sets and Systems*, 64(1), pp. 21-30.

Resnick, P, Iacovou, N, Sushak, M, Bergstrom, P, & Riedl, J (1994) ‘GroupLens: An Open Architecture for Collaborative Filtering of Netnews.’, *Proceedings of ACM 1994 Conference on Computer Supported Collaborative Work Conference*, pp. 175-186.

Rocchio, JJ (1971), ‘Relevance Feedback in Information Retrieval’, *In the SMART Retrieval System: Experiments in Automatic Document Processing (ed. Salton, G)*, pp.313-323. Prentice Hall International, Inc.

Rodden, T (December 1991), ‘A survey of CSCW Systems’, *Interacting with Computers,* 3(3), pp.319-353.

Salton, G & Buckley, C (1988), ‘Term Weighting Approaches in Automatic Text Retrieval’, *Information Processing & Management,* 24(5), pp. 513-523.

Schafer, B, Konstan, J & Riedl, J (2000), ‘E-Commerce Recommendation Application’, *Journal of Data Mining and Knowledge Discovery*, 5(1), pp. 115-152

Wiig, K (1993), *Knowledge Management Foundation.* Schema Press.

Yager RR (1988) ‘An Ordered Weighted Averaging Operators in Multicriteria Decision Making’, *IEEE Trans. on Syst., Man and Cybern. B,* 18(1), pp. 183-190.

Zadeh, LA (1965) ‘Fuzzy sets’, *Information and Control,* 8(3), pp. 338-353.

Zadeh, LA (1975), ‘The Concept of a Linguistic Variable and its Application to Approximate Reasoning, Parts 1, 2, and 3’, *Information Sciences,* 8(2), pp. 199-249; 8(3), pp. 301-357; 9(1), pp. 43-80.

Appendix I

Table 7 and Table 8 show the illustrated examples of the task set and categories selected from a research department. Categories are defined according to the features of the task set based on the schema of ACM Computing Classification Systems (1998) http://www.acm.org/class/ 1998/

Table 7. Illustrated Examples of Categories

Category ID	Category Name
1	Database and Management
2	Information System Applications
3	Management of Computing and Information Systems
4	Computers and Society

Table 8. Illustrated Examples of Tasks

Task ID	Task Name
1	Classifying Video Data
2	Integrating Healthcare EDI and SET
3	Modeling XML-based Workflows
4	ERP Systems
5	Information Filtering in E-Catalogs
6	Multidimensional Transaction Analysis
7	E-Catalogs Analysis
8	Deploying Data Warehouse
9	Web Logs Analysis
10	Designing Composite E-service