

Toward incorporating a task-stage identification technique into the long-term document support process

I-Chin Wu^{a,*}, Duen-Ren Liu^b, Pei-Cheng Chang^b

^a *Department of Information Management, Fu Jen Catholic University, Taipei 242, Taiwan*

^b *Institute of Information Management, National Chiao Tung University, Hsinchu 300, Taiwan*

Received 27 June 2007; received in revised form 22 November 2007; accepted 25 November 2007

Available online 22 January 2008

Abstract

Effective knowledge management in a knowledge-intensive environment can place heavy demands on the information filtering (IF) strategies used to model workers' long-term task-needs. Because of the growing complexity of knowledge-intensive work tasks, a profiling technique is needed to deliver task-relevant documents to workers. In this study, we propose an *IF technique with task-stage identification* that provides effective codification-based support throughout the execution of a task. Task-needs pattern similarity analysis based on a correlation value is used to identify a worker's task-stage (the pre-focus, focus formulation, or post-focus task-stage). The identified task-stage is then incorporated into a profile adaptation process to generate the worker's current task profile. The results of a pilot study conducted in a research institute confirm that there is a low or negative correlation between search sessions and transactions in the pre-focus task-stage, whereas there is at least a moderate correlation between search sessions/transactions in the post-focus stage. Compared with the *traditional IF technique*, the proposed *IF technique with task-stage identification* achieves, on average, a 19.49% improvement in task-relevant document support. The results confirm the effectiveness of the proposed method for knowledge-intensive work tasks.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Correlation analysis; Information filtering; Knowledge-intensive task; Long-term task performance; Task profile; Task-stage identification

1. Introduction

Intellectual content containing valuable knowledge is usually codified in an explicit form to facilitate reuse (Bolloju, Khalifa, & Turban, 2002; Markus, 2001; Nonaka, 1994). A repository of structured and explicit knowledge, especially in document form, is a codified strategy for managing knowledge (Davenport & Prusak, 1998; Zack, 1999). Empirical findings indicate that codifying intellectual content into a knowledge repository

* Corresponding author.

E-mail address: icwu.fju@gmail.com (I-Chin Wu).

helps workers exploit existing organizational resources effectively (Gray, 2001). However, with the growing amount of information in organizational memories, as well as the increasing complexity of executing tasks in a knowledge-intensive working environment, knowledge workers are suffering increasingly from information overload.

Information filtering with a similarity-based approach is often used to locate knowledge items relevant to the task-at-hand. The relevance of a knowledge item is determined by analyzing the similarity between the concept terms of the item and the current task (Baeza-Yates & Ribeiro-Neto, 1999; Holz, Maus, Bernardi, & Rostanin, 2005; Ye & Fischer, 2002). Holz et al. proposed a task-oriented, similarity-based approach that organizes desktop documents and proactively delivers task-specific information. The effectiveness of this approach depends on the accuracy of the concept terms of the task-at-hand, which are extracted from the task name and associated relevant documents. However, because of the nature of knowledge-intensive work tasks, such as research projects in academic institutions and product development in R&D departments, it is difficult to provide task-relevant knowledge based solely on static descriptions of the information, e.g., the task's title, goal, and participants, during a task's execution. This is because a knowledge-intensive work task usually consists of levels of progressively smaller sub-tasks, so it can take a long time to complete the task. In addition, workers' task-needs usually vary according to the task-stage of the long-term project. Therefore, providing knowledge to meet those task-needs is a long-term knowledge support issue.

IF systems are often personalized to support the long-term information needs of a particular user or a group of users (Mostafa, Mukhopadhyay, Lam, & Palakal, 1997; Widyantoro, Ioerger, & Yen, 2001). Various methods for learning users' interests or preferences from text documents or web pages have been proposed for real-world applications based on recommender systems, for example, adaptive web page recommendation services (Balabanović, 1997), e-mail-filtering systems (Mostafa et al., 1997), personalized online newspapers (Billsus & Pazzani, 1997), and on-line recommendation of academic research papers (Middleton, Shadbolt, & De Roure, 2004). These approaches rely on learning algorithms that modify users' profiles to reflect users' interests or preferences. In Information Retrieval or Information Theory, several well-known methods are employed to model users' constantly changing information needs; for example, Rocchio's algorithm, information gain theory, Bayesian classifiers, and the *k*-Nearest Neighbor algorithm. The above learning techniques incorporate relevance feedback mechanisms that learn workers' dynamic information needs based on explicit feedback (where the system collects users' linguistic ratings) or implicit feedback (where the system monitors users' access behavior). Contemporary IF systems that rely on user models (e.g., user profiles) provide an effective means of modeling a worker's dynamic information needs based on explicit or implicit relevance feedback techniques (Kelly, 2004; Ruthven, Lalmas, & van Rijsbergen, 2003; White & Kelly, 2006; Widyantoro et al., 2001). However, as mentioned previously, such systems are limited because they cannot fully reflect the workers' current task-needs. Thus, a knowledge support model that considers the specific task-stage and the worker's explicit and implicit task-needs should be incorporated into the traditional IF model.

Several empirical studies have analyzed the growth in students' or scholars' understanding of their assigned tasks while conducting an actual research project (Kuhlthau, 1993; Vakkari, 2000; Vakkari, Pennanen, & Serola, 2003; Wang & Soergel, 1998; Wang & White, 1999). The Vakkari studies, which focus on a user's information seeking activities during a task's performance (e.g., writing a proposal, or executing a project), show that information needs vary according to different task stages (Vakkari, 2000). Vakkari classified the stages of the search process according to Kuhlthau's (1993) search process model, namely, the *pre-focus*, *focus formulation*, and *post-focus stages*. In each stage, the type of information needed may vary from general to specific data, and the choice of search terms may range from broad to specific terms. The stage features defined by previous pilot studies allow us to identify a worker's task stages (Kuhlthau, 1993; Vakkari, 2000). The importance of identifying stages or milestones has gained increasing attention in the KMS field in recent years. For example, Ahn, Lee, Cho, and Park (2005) presented a virtual workgroup support system (VWSS) combined with a knowledge context model (KC-V) to support a project's (task's) execution. The concept of the evolutionary accumulation of knowledge combined with collaborative activities during the long period of a project's execution forms the basis of the KC-V model. Each project has several decision points, i.e., milestones or project status points, which are defined so that task-associated knowledge items can be stored at specific points of the project.

In a previous work, we proposed a knowledge support model for a task-based working environment (Liu, Wu, & Yang, 2005). Based on that model, we have developed a digitalized task-based workplace. Preliminary investigations demonstrate the effectiveness of the proposed model and workplace. In this paper, we propose an *IF technique with task-stage identification* that extends the traditional IF technique by adding the worker's task-stage. Notably, our *task* definition is closer to a genuine work-task, which requires *problem and task solving knowledge* and *search task solving knowledge* to execute complex, long-term tasks (Ingwersen & Järvelin, 2005). That is, the worker knows the seeking and retrieval strategies and methods, and he/she requires task and domain information acquisition for long-term task performance. Therefore, we observed the worker's search behavior, i.e., a series of explicit and implicit feedback behavior patterns related to documents needed to complete the task at hand. In this work, we integrate a task-stage identification technique with a long-term document support process to fit the worker's task-needs more precisely. To evaluate the effectiveness of the proposed knowledge support model, we conducted a 1-year empirical investigation of its application in our problem domain. The study showed that a worker has different information needs as a task progresses, which confirms the observations in the Vakkari studies. Each stage has certain characteristics and patterns that enable us to identify the worker's task needs. Thus, we propose a task-stage knowledge support model that provides efficient knowledge support by identifying a worker's current task-stage and task-relevant topics. We also propose a method for analyzing the similarity of task-need patterns. A correlation value is used to identify a worker's task-stage, and a task-based domain ontology, i.e., a topic taxonomy, is employed for task-relevant topic identification and pattern analysis. The task-related topic taxonomy in our application domain is a multi-level structure in which each node represents a research topic. Thus, based on the topic taxonomy, a worker's access behavior and changing search patterns can be analyzed. Correlation analysis is then used to identify the change points of stages according to the results of pattern analysis.

The remainder of the paper is organized as follows. Section 2 contains a literature review. Section 3 gives an overview of the task-stage knowledge support model. Section 4 provides details of usage pattern modeling, i.e., IF with a task-stage identification technique. The method of document support based on the worker's task-stage is also described in this section. The experiment's objectives, procedures, results, and implications are given in Sections 5 and 6. Finally, in Section 7, we present our conclusions and discuss the direction of our future work.

2. Literature reviews

In this section, we discuss related work and the contributions of the present study.

2.1. Task-based knowledge management

Historical codified knowledge, i.e., experiences and know-how extracted from previously executed tasks, provides valuable knowledge that can help in the performance of new tasks. For example, effective project management can benefit from KMS by referring to similar past projects to obtain information about best practices, lessons learned, and knowledge resources. In the research field, task innovation is generally based on previous research achievements.

In recent years, task-based knowledge management has emphasized the organization of all possible task-specific knowledge by supporting context-aware knowledge access and retrieval (Abecker, Bernardi, Hinkelmann, Kühn, & Sintek, 2000; Alvarado, Romero-Salcedo, & Sheremetov, 2004; Celentano, Fugini, & Pozzi, 1995; Kwan & Balasubramanian, 2003). Information retrieval (IR) techniques coupled with workflow management systems (WFMS) are used to support proactive delivery of task-specific knowledge according to the context of tasks in a process (Abecker et al., 2000; Fenstermacher, 2002). Furthermore, a process meta-model that places the knowledge in context is integrated into workflow systems to capture and retrieve knowledge within a process context (Kwan & Balasubramanian, 2003). Alvarado et al. (2004) proposed organizing corporate memory from the perspective of role/job positions, whereby organizational memory, modeled by adopting UML/XML, is used to specify ontologies for the organization's positions, tasks, and application domains. Although there are subtle differences between the above works, overall, they

provide an appropriate perspective because they specify the process-context of a task to support context-aware knowledge retrieval. That is, context is seen as an important component that can be utilized to improve the understanding of relevant knowledge about a business task within the KMS. However, without a proper model of the associated process to capture the specific work-task context, it is difficult to achieve effective knowledge management. Additionally, it is hard to use simple and static processes to describe all aspects of knowledge-intensive tasks.

Task-based knowledge delivery combined with an information filtering technique is well-suited to knowledge-intensive tasks because of its ability to model worker's dynamic task-needs. Furthermore, a profiling technique is appropriate for knowledge-intensive and non-routine tasks because of its learning capability. In this type of framework, information filtering techniques and an intelligent agent-based architecture are usually adopted to streamline knowledge delivery from internal and external knowledge repositories (Spies, Clayton, & Noormohammadian, 2005; Ye & Fischer, 2002). De Bra et al. (1997) proposed a cooperative agent architecture to achieve task-based information filtering within the work process. In this model, three types of cooperating agents: process agents, document warehouse agents, and retrieval agents are used to determine if retrieved documents are relevant to the worker's task at hand. Subsequently, a *CodeBroker* system was proposed to facilitate the reuse of software in the organization's repository (Ye & Fischer, 2002). In summary, the above approaches provide knowledge support by exploiting a task's characteristics and employing user modeling techniques.

2.2. Long-term document support for knowledge-intensive work tasks

Information filtering (IF) systems are usually personalized to support the long-term information needs of a particular user or a group of users (Mostafa et al., 1997; Widiantoro et al., 2001). IF systems are similar to conventional information retrieval (IR) systems that support a user's short-term information needs, e.g., general information needs expressed in a single search session. However, although the IF system relies on the support of the kernel technology of IR, it emphasizes methods that learn and maintain user profiles to support long-term information services (Baeza-Yates & Ribeiro-Neto, 1999; Belkin & Croft, 1992; Widiantoro et al., 2001). IF techniques have been incorporated into many real-world applications (Balabanovič, 1997; Billsus & Pazzani, 1997; Middleton et al., 2004; Mostafa et al., 1997) and filtering technology is acknowledged as an effective way to reduce information overload and provide personalized information (Hanani, Shapira, & Shoval, 2001; Hoashi, Matsumoto, Inoue, & Hashimoto, 2000). Although IF systems provide proper profiling methods to learn a user's dynamic information needs and interests, most existing systems do not integrate those needs with the characteristics of the relevant task-stage as the execution of the task progresses.

Generally, a worker uses documents to understand a task and solve a specific problem. Thus, when a worker begins a task, he/she may search the organization's knowledge repository for information that will help solve the problem at hand. The worker's search behavior results from the fact that there is a gap between his/her knowledge about the task and the perceived requirements of the task. The gap is called the information need and results in information seeking activities (Byström & Järvelin, 1995), i.e., a series of information retrieval activities. Several empirical studies have focused on discovering and analyzing the growth in students' or scholars' understanding of their assigned tasks during actual research projects (Kuhlthau, 1993; Vakkari, 2000; Vakkari et al., 2003; Wang & Soergel, 1998; Wang & White, 1999). These works analyze how documents are selected and used by workers during a task's execution. Kuhlthau's (1993) search process model is a longitudinal study of library users seeking information for the completion of an assigned project. Vakkari's model, which is based on Kuhlthau's search process model, classifies researchers' information seeking activities into pre-focus, focus-forming, and post-focus stages. Both studies show that a worker's information needs and information-seeking behavior depend on the progress of the task, or task stages. In addition, the concept of an exploratory search system to support online searches has been widely discussed in recent years (White, Kules, Drucker, & Schraefel, 2006; White & Kelly, 2006). The goal of the system is to help users find specific knowledge from the tremendous amount of data available on the Web. A user may only have a general idea about a topic and may be uncertain about what information is required for the task at hand. (Jansen, 2005; White, Jose, & Ruthven, 2003; White & Kelly, 2006).

In this work, the concept that information seeking behavior is related to task stages is adopted to enhance traditional IF techniques and provide effective knowledge support. That is, we try to identify the worker's task-stage and incorporate the result into the profile modeling process so that the system can deliver task-relevant documents, and thereby provide effective long-term knowledge support.

3. Overview of task-stage knowledge support study

Because workers have dynamic information needs in different task stages, we propose a method that can identify those needs. Section 3.1 introduces the main modules that streamline the operation of task-stage knowledge support, and Section 3.2 describes the task-based domain ontology, i.e., the task-based topic taxonomy.

3.1. Main modules of task-stage knowledge support

As Fig. 1 illustrates, providing relevant task-stage knowledge involves three phases: *data pre-processing*, *task-needs discovery*, and *adaptive task-stage knowledge support*. Note that the task-needs discovery phase is the kernel of the system because it analyzes the worker's task-relevant topics and task-stages.

Two types of valuable information, content data and usage data, are acquired during the *data pre-processing phase*. The text pre-processing module extracts information from unstructured or semi-structured data. The *user behavior tracker* is an on-line module that tracks a user's interaction with the system. The user's task-related behavior, including access to the task-based domain ontology and relevance feedback on knowledge items, can be captured and recorded in his/her profile. The *profile handler* uses an adaptive task-based profiling approach to adjust workers' profiles based on their dynamic behavior. The *task-needs discovery phase* is the kernel of the system. In the task-needs discovery phase, the *task-stage identifier* determines the worker's task-stage based on a variety of profiles, while the *task-relevant topics analyzer* analyzes task-relevant topics:

- *Task-stage identifier*: The task-stage identifier is responsible for determining and analyzing the worker's task-stage based on changes in the task-needs pattern over time.

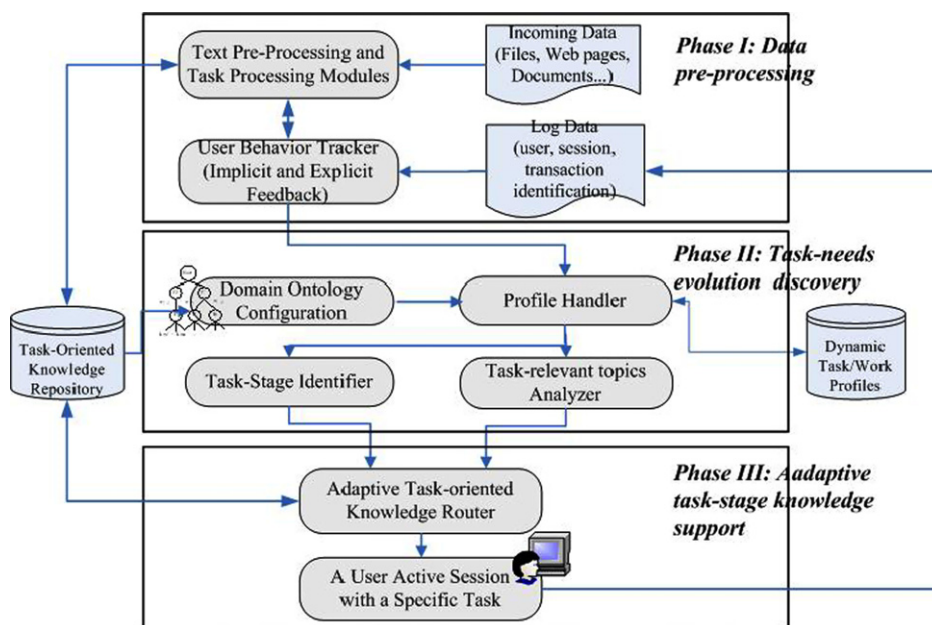


Fig. 1. Task-stage knowledge support.

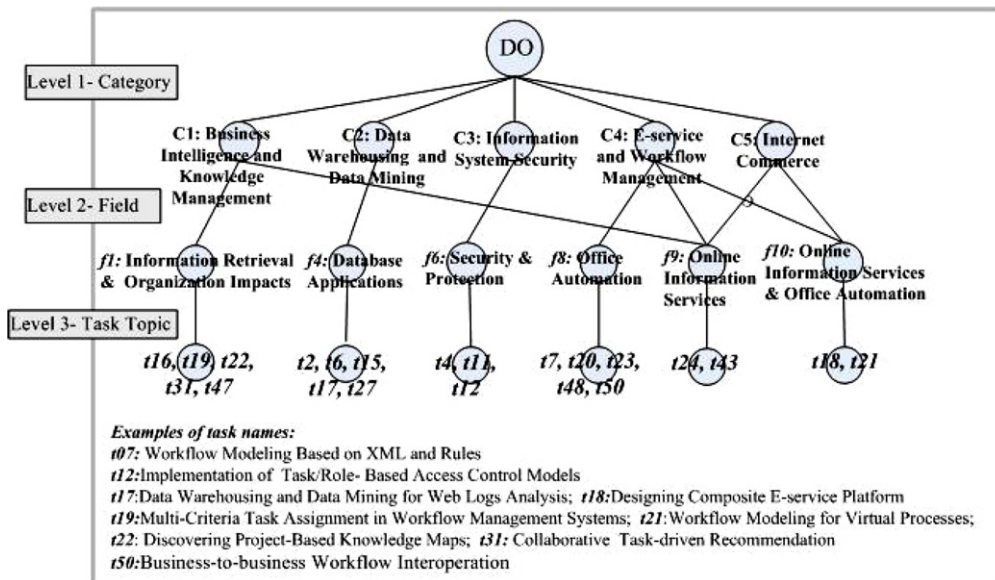


Fig. 2. An example of topicalities in the task-based domain ontology.

- *Task-relevant topics analyzer*: This mechanism determines the worker's task-needs in terms of topics in the domain ontology (DO), as shown in Fig. 2, by tracking the worker's access behavior over a period of time. The behavior is then analyzed based on the DO to discover task-relevant topics for the worker. Accordingly, a worker's current task-needs can be expressed as a set of topics with their degrees of relevance, i.e., a task-needs pattern.

We employ a profiling technique to model a worker's task-needs for a specific task. Profiling has been used in recommender systems and KMSs to enhance knowledge access and promote knowledge sharing among peer- or project-based groups (Abecker et al., 2000; Alvarado et al., 2004; Balabanovi'c, 1997; Belkin & Croft, 1992). As a result, the IF strategy can provide stage-relevant knowledge based on the analysis in the task-needs discovery phase. The adaptive task-oriented knowledge router provides workers with pertinent task-relevant codified knowledge based on his/her task profile and task-stage.

3.2. Task-based domain ontology

The domain ontology (DO), which is a shared conceptualization of a specific domain, is often used to specify an organization's working domain (Park & Hunting, 2003). Organizing knowledge items into an ontological structure based on the domain ontology is an effective way to support knowledge retrieval in business environments. In this work, we consider the DO as a classification structure of tasks stored in the knowledge repository. Accordingly, the proposed task-based DO is a simple topic taxonomy that is structured in four levels: categories, fields, tasks, and knowledge items, as shown in Fig. 2. The task-based DO is used to organize codified knowledge around specific organizational tasks.

Specifically, in this work, we refer to each node in the task level of the proposed DO as a topic i.e., there a one-to-one relationship between a task and a topic. Thus, a topic is also referred to as a task topic. Each task topic has a task corpus (topic profile) that is extracted from set of task-related documents. The corpus describes the key features of tasks and can be constructed by extracting key content from textual data gathered during a task's execution. More details can be obtained from our previous work (Liu et al., 2005). The task-based DO is used to organize codified knowledge around specific organizational tasks. In addition, the worker's current task needs can be expressed as a feature vector of weighted keywords and a set of topics with

associated weights. The relevance of topics is shown in the field or task level of the DO. Obviously, a task topic is an abstract representation of an executed task, i.e., a historical, completed task. Accordingly, we use the task name to represent the topic in this work, as shown in [Appendix B](#). In this work, we focus on changes in a worker's behavior in the task-level; therefore, we only consider topics in the task level.

4. Data preprocessing and task-needs discovery

The data pre-processing phase, which is the most important and time-consuming step of the knowledge discovery process, is discussed in [Section 4.1](#). [Section 4.2](#) describes the modeling of a worker's usage pattern based on the analysis of a variety of task profiles. In [Section 4.3](#), we propose a task-needs pattern similarity analysis method based on a correlation value to identify changes in a worker's task stages. Then, the profile adaptation method based on the task-stages is described in [Section 4.4](#).

4.1. Data pre-processing

In this sub-section, we consider two valuable information sources for knowledge acquisition: content data and usage data.

4.1.1. Content data

Information indexed by appropriate concepts and categories can provide knowledge workers with meaningful access to their organization's intellectual content. Rather than simply adopt a document index server to provide document retrieval services, the proposed model analyzes and organizes an organization's previous tasks (existing tasks) from the perspective of each previous task unit. The *text pre-processing module* is responsible for term transformation and term weighting tasks. The key content of a codified knowledge item (document) is represented as a feature vector of weighted terms in n -dimensional space, using a term weighting approach that considers term frequency, inverse document frequency, and normalization factors ([Salton & Buckley, 1988](#)). The *term transformation* steps, i.e., case folding, stemming, and stop word removal, are performed during text pre-processing ([Porter, 1980](#); [Witten, Moffat, & Bell, 1999](#)). Then, *term weighting* is employed to extract the most discriminating terms. Thus, each document d_j is pre-processed and represented as a feature vector \vec{d}_j . Next, two off-line batch-processing units in the *task processing module*, a task corpus selection engine and a task categorization engine, deal with knowledge items for each task. A centroid approach is used to derive the feature vector of a task by averaging the feature vectors of documents generated/accessed during the search. Let D_{t_r} denote the set of documents generated/accessed by task t_r . The task corpus (feature vector) of task t_r is defined as the *centroid* vector \vec{t}_r , obtained by averaging the feature vectors of documents in D_{t_r} . [Eq. \(1\)](#) defines the *centroid* vector \vec{t}_r . The weight of a term k_i in \vec{t}_r is represented by $w(k_i, t_r)$:

$$\vec{t}_r = \frac{1}{|D_{t_r}|} \sum_{d_j \in D_{t_r}} \vec{d}_j \quad (1)$$

An associated task-based repository that preserves the content and structure of knowledge is established to store task-relevant data in the database.

4.1.2. Usage data

The user behavior tracker is an on-line module that captures workers' information seeking behavior. A *task transaction*, denoted by T_{trans} , is used to analyze a worker's implicit and explicit feedback behavior. The log-parsing engine in the user behavior tracker analyzes log-files that track the user's interaction with the system. Each log record is analyzed to extract important information, such as the user's name, request type, docID and the time of the request. The user's login data, such as his position and TaskID, is combined with the log record and stored in the log database. In the database, each record is represented as a record (user name, -role type, TaskID, DocID, request type, system time). In summary, the log-parsing engine collects and manages task-relevant information by tracking a user's document access behavior for a specific task.

4.2. Usage pattern modeling

The worker’s temporal task-needs, i.e., information needs for a task over a period of time, are expressed as a temporal task profile and represented by a feature vector of weighted keywords. In addition, the task-relevance analyzer determines the degree of relevance between the task profile and the topics profile in the task-based domain ontology (DO). Thus, a worker’s temporal task-needs pattern can be expressed as a set of topics and associated relevance degrees.

Next, we define a *session* and a *task transaction*, which are used to analyze workers’ implicit and explicit feedback on codified knowledge items. The user behavior tracker performs the analysis by monitoring the worker’s access behavior over a period of time. Fig. 3 shows a snapshot of the monitoring process:

- A *session* is defined as a sequence of user feedback actions related to knowledge items (e.g., reading, downloading, or rating an item of information) during a single visit to the system.
- A *task transaction* records the worker’s access to the knowledge repository over a number of sessions. The time interval between two consecutive upload events (i.e., documents the worker wants to share in the presented system) is defined as the task-transaction in our investigated domain. In the context of our research domain, a worker who uploads documents indicates that he/she has recently explored the related topics and will give a presentation in the laboratory’s regular meeting. A session is a smaller unit than a *task transaction*. During the period of a task transaction, the worker also engages in some search behavior in the proposed system, such as reading and rating articles. Accordingly, a worker’s task *transaction* comprises n sessions, where $n > 0$.

4.2.1. Feedback analysis

The temporal profile of a worker’s sessions or transactions is generated according to the tracking results of his/her sessions or transactions over a period of time. In addition, the system analyzes the relevance of task topics in the domain ontology (DO) to the session/transaction temporal profile. Consequently, a worker’s task-needs pattern is expressed as a set of topics that are task-level nodes in the DO. We use a real example to explain how to detect and track a worker’s access behavior and conduct task-needs pattern modeling.

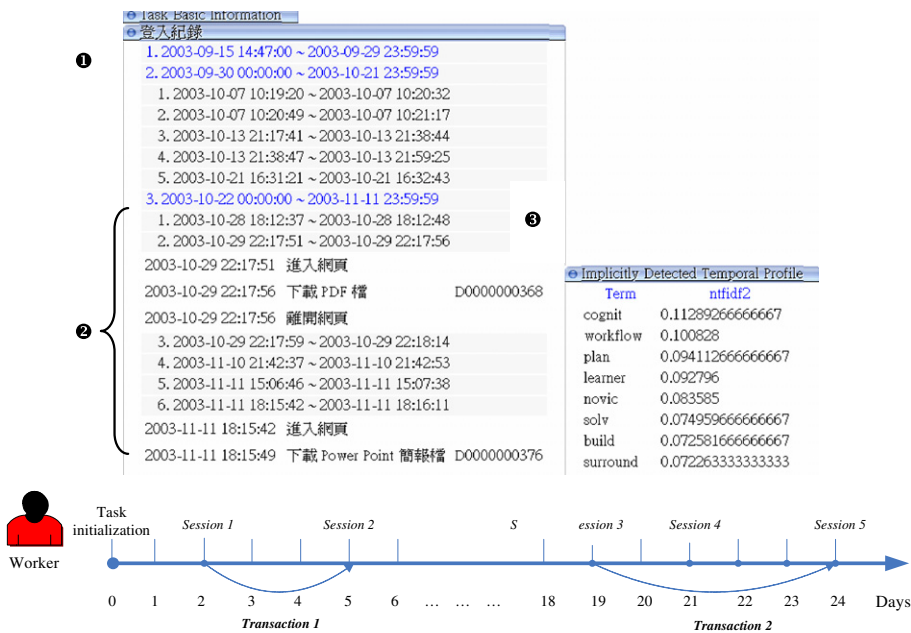


Fig. 3. Snapshot of the feedback behavior tracking system.

4.2.2. Example 1 (ref. Fig. 3)

In this example, six sessions are identified in the third transaction of the executor “*P.T. Chou*”, the executor of the “*ITIL-based Context-aware Knowledge Recommendation*” task. The task is a research project. As mentioned previously, in our research application domain, the time interval between two consecutive upload events is defined as a *task transaction*. In the given example, the third transaction comprises six sessions. The transaction started from 2003/10/22 and ended at 2003/11/11. After conducting data preprocessing in phase one, a set of the worker’s access behavior patterns across sessions, $Trans_i = \{s_1, s_2, \dots, s_n\}$, and a set of accessed knowledge items, $O = \{I_1, I_2, \dots, I_m\}$, are defined. Herein, knowledge items denote the documents or Power-Point files. The blue color in Fig. 3 denotes a $Trans_i$ (as denoted by circle one), whereas the set of time interval below the $Trans_i$ are *sessions* which records the time when the *P.T. Chou* log into the system (as denoted by circle two). As we have defined, a transaction comprises n sessions, where $n \geq 0$. Notably, the circle three in Fig. 3 denotes the *P.T. Chou* accessed a document, D0000000368 (i.e., I_{368} for expression shortly), during the Session 2 of Transaction 3. Furthermore, he accessed the $I_{376}, I_{458}, I_{376}, I_{375}, I_{460}$ in the Session 6 of Transaction 3 (i.e., partially shown in the Fig. 3). The right hand side of Fig. 3 denotes the temporal profile of Session 2 in the Transaction 3, i.e. set of stemmed keywords with associated weights. Accordingly, a sequence of knowledge items accessed in each session of a transaction, i , are identified as follows:

$$AI(Trans_3^{S2}) : \langle I_{368} \rangle$$

$$AI(Trans_3^{S3}) : \langle I_{376}, I_{458}, I_{376}, I_{375}, I_{460} \rangle$$

$$AI(Trans_3^{S6}) : \langle I_{376}, I_{461}, I_{362}, I_{375}, I_{368}, I_{376} \rangle$$

$AI(Trans_i^{S_l})$ represents a sequence of knowledge items accessed in a session, l , of a transaction, i . The user’s access behavior pattern includes implicit feedback behavior and explicit feedback behavior. In this work, implicit feedback behavior refers to instances when the user “uploaded or download documents” or “read the documents on-line”, whereas explicit feedback behavior refers to when the user “rated the documents”. In this work, explicit feedback is given by a *linguistic rating*, which is “Perfect”, “Very High”, “High”, “Normal”, “Low”, and “Very Low” via the feedback bar provided in the proposed system. The above pattern of the given example indicates that the implicit and explicit behavior related to knowledge item I_{376} occurs at different times in the same session.

4.2.3. Temporal profile generation

The temporal profile is derived from the *feature* vectors of documents accessed by a worker, u , over a specific time period, as shown in the Eq. (2); $Trans_i^{S_l}$ denotes the temporal profile (feature vector of weighted terms) derived from the documents accessed in session l of transaction i . The profile is generated by the *profile handler* to represent a worker’s current information needs for the target task:

$$\overrightarrow{Trans_i^{S_l}} = \frac{1}{|D_{i,s_l}^{exp}|} \sum_{\forall d_j \in D_{i,s_l}^{exp}} (A(d_j) \times \vec{d}_j) + \frac{1}{|D_{i,s_l}^{imp}|} \sum_{\forall d_j \in D_{i,s_l}^{imp}} (CV(\tilde{H})^u \times \vec{d}_j), \quad (2)$$

where D_{i,s_l}^{exp} denotes the set of documents rated explicitly by a worker for the target task during session l of transaction i ; and $A(d_j)$ is derived from a worker’s linguistic ratings on the relevance of document d_j to the target task and then transformed into a crisp value. The transformation process follows the fuzzy linguistic approach in order to approximate human perception and provide easier assess to qualitative problems (Zadeh, 1975). The reason is that we found the linguistic rating is a more natural and therefore easier method of relevance assessment for knowledge workers, while the fuzzy number is used in the back-end of the system to facilitate numerical computation of the relevance ratings. D_{i,s_l}^{imp} denotes the set of documents uploaded, downloaded or read, but not rated, by the worker during session l of transaction i . A linguistic rating “High” is given by default to represent the relevance degree of unrated documents (implicit feedback). $CV(\tilde{H})^u$ denotes a “High relevance rating” of a worker u and is also transformed into a corresponding crisp value. Since our system shows the description of a document, we assume that a worker will read the description first to decide if the document is relevant, and then read the document in detail.

4.2.4. Task-need patterns

The *task-relevant topic analyzer* calculates the similarity (e.g., the cosine measure) between the temporal profile $Trans_i^{S_l}$ and the profile of a topic $\overrightarrow{topic_j}$, which represents the associated profile (feature vector of weighted terms) of $topic_j$. Note that topic j also represents a research topic, which is the task node in the proposed domain ontology (DO). Accordingly, the pattern of a worker’s task needs can be expressed as a set of topics and associated relevance degrees, as shown in Table 1. The task-needs pattern of a session l in transaction i , denoted by $Patt_{Trans_i}^{S_l}$, is expressed as a set of topics with their associated relevance degrees ($topic_j, rd_j$). Notably, Appendix B lists topics in the domain.

4.3. Task-stage identification based on usage pattern modeling

In this section, we describe a *task-needs pattern similarity analysis method* based on the correlation value to identify changes in a worker’s task-stages. The objective of task-stage identification is to deliver task-relevant knowledge according to the worker’s task-needs based on different task-stages. From previous pilot studies (Kuhlthau, 1993; Vakkari, 2000), we identify three task stages: the pre-focus, focus formulation, and post-focus_stages, which differentiate a worker’s three types of information needs during a task’s execution.

4.3.1. Stage identification process

The on-line task-stage identifier analyzes and determines a worker’s task-stage based on his/her access pattern. The *task temporal profile* in each timeframe is the basis for identifying the worker’s task-stage. A change of task-stage is inferred by analyzing the correlation of the task’s temporal profile of the worker’s consecutive transactions. Three steps are executed to analyze and determine the worker’s task-stage:

Step 1. Task-need pattern calculation: As noted in Section 4.2, a worker’s task-need pattern can be expressed as a set of topics and associated relevance degrees. The task-need pattern of a session l in transaction i , $Patt_{Trans_i}^{S_l}$, is expressed as a set of topics with their associated relevance degrees ($topic_j, rd_j$). The relevance degree of a topic j , rd_j , is derived by the cosine measure $sim(Trans_i^{S_l}, \overrightarrow{topic_j})$. The cosine measure of feature vectors is used as the similarity measure.

Similarity measure: The cosine formula is a similarity measure that is widely used to assess the degree of similarity between two items, x and y . It computes the cosine of the angle between the items’ corresponding feature vectors \vec{x} and \vec{y} , as shown in Eq. (3). The degree of similarity is higher if the cosine similarity is close to 1.0:

$$sim(x, y) = cosine(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} \tag{3}$$

$\overrightarrow{Trans_i^{S_l}}$ denotes the temporal profile (feature vector of weighted terms) derived from the documents accessed in session l of transaction i . Table 1 shows an example of $Patt_{Trans_i}^{S_l}$ and $Patt_{Trans_i}$. The latter is defined and generated in a similar manner by considering the whole transaction.

Step 2. Correlation calculation: Once the task-needs pattern has been derived, the correlation of the worker’s task-needs pattern across transactions can be calculated by *Pearson’s correlation coefficient*. It is reasonable to assume that a worker’s task-needs will not change dramatically during consecutive sessions of the same transaction, but they may change significantly between two different transactions. Thus, we calculate the correlation between the previous transaction, $Trans_{i-1}$, and the start session of the current transaction, $Trans_i^{S_1}$, as shown in Eq. (4):

Table 1
Correlation of task-needs patterns

	T_1	T_2	T_3	T_4	...	T_8	...	T_{12}	T_{13}	T_{14}	...	T_{19}	T_{20}	T_{21}	T_{22}	T_{23}	...	T_{26}	...	T_{28}	T_{29}	T_{30}	...	T_{34}	T_{35}
$Patt_{u, Trans_i}$	0.11	0.56	0.11	0.06	...	0.05	...	0.26	0.09	0.07	...	0.10	0.08	0.10	0.11	0.10	...	0.08	...	0.12	0.12	0.13	...	0.08	0.31
$Patt_{u, Trans_{i+1}S_1}$	0.04	0.05	0.03	0.02		0.02		0.06	0.05	0.13		0.01	0.03	0.05	0.09	0.10		0.06		0.15	0.09	0.07		0.03	0.05

$corr(Patt_{Trans_{i-1}}, Patt_{Trans_i}^{S_1}) = -0.024$. The correlation between two transactions is “Low”.

$$corr_u(A, B) = \frac{\sum_{j \in \text{topicset}} (rd_j^A - \overline{rd^A})(rd_j^B - \overline{rd^B})}{\sqrt{\sum_{j \in \text{topicset}} (rd_j^A - \overline{rd^A})^2 \sum_{j \in \text{topicset}} (rd_j^B - \overline{rd^B})^2}} \quad (4)$$

Let A represent $Patt_{Trans_{i-1}}$ and B represent $Patt_{Trans_i}^{s_1}$; rd_j^A and rd_j^B are the relevance degrees of topic j in $Patt_{Trans_{i-1}}$ and $Patt_{Trans_i}^{s_1}$, respectively; $\overline{rd^A}$ and $\overline{rd^B}$ are the average relevance degrees of the topics in $Patt_{Trans_{i-1}}$ and $Patt_{Trans_i}^{s_1}$, respectively.

Step 3. Changes in task stages: Changes in a worker’s task stages are based on the correlation values between task-needs patterns. The rationale behind the proposed *correlation analysis method* is that we can identify a worker’s task-stage based on the changes in the task-needs for topics, i.e., the task-need pattern in our work. In other words, some task-relevant topics in the task-based topics taxonomy may have a high degree of relevance to the temporal profile of the previous transaction; however, they may have a low degree of relevance to the temporal profile at the beginning of the current transaction. Because the correlation values are within the range $[-1, 1]$, it is easy to track the worker’s access pattern based on the correlation value between transactions. Using a correlation analysis method, we took around one year to observe the behavior of workers when they accessed the knowledge repository in the presented task-based workspace. Based on the results of our sample analysis, we set a three-part correlation range: “Low”, “Moderate”, and “High”.

Low correlation: If a worker is in the early stages of executing a task, the correlation value between transactions will be within the interval $[-1, 0.2]$, which indicates that he/she is in the task pre-focus stage and is uncertain about the perceived task.

Moderate correlation: If a worker has decided the research area, but feels uncertain about the research topic of a specific area, the correlation value between transactions will be within the interval $[0.2, 0.6]$, which indicates that he/she is approaching or is in the task focus formulation stage.

High correlation: Once a worker has focused on a specific topic, the correlation value between transactions will be within the interval $[0.6, 1.0]$, which indicates the worker has defined his/her task-needs for a specific topic. In other words, the worker accessed and read documents belonging to the related topics. Hence, the worker is in the task post-focus stage.

The time point for deciding the worker’s task-stage is the first session of the current transaction, $Trans_i$. Assuming we know the worker’s task-stage in the previous transaction, $Trans_{i-1}$, we can infer his/her current task-stage based on the correlation value and the worker’s previous task-stage. Table 2 shows the task-stage determination rule.

4.3.2. Sample analysis

Example 2: There are 35 task-level topics (historical task corpora) in our domain, as shown in Appendix B. Continuing with Example 1, we analyze the four transactions in the access behavior of the executor “PoTsun”.

Table 2

Task-stage determination rule

Input: $corr_u(Patt_{Trans_{i-1}}, Patt_{Trans_i}^{s_1})$: Correlation values between transactions
$Trans_{i-1}$ · stage: Task-stage of $i-1$ th transaction (previous transaction)
Output: $Trans_i$ · stage: Task-stage of i th transaction (current transaction)
Case of $Trans_{i-1}$ · stage
“pre-focus stage”: If $corr_u(Patt_{Trans_{i-1}}, Patt_{Trans_i}^{s_1})$ is “low”
$Trans_i$ · stage = task pre-focus stage
Else $Trans_i$ · stage = task formulation stage;
“formulation stage”: If $corr_u(Patt_{Trans_{i-1}}, Patt_{Trans_i}^{s_1})$ is “high”
$Trans_i$ · stage = task post-focus stage
Else $Trans_i$ · stage = task formulation stage;
“post-focus stage”: $Trans_i$ · stage = task post-focus stage;
Return ($Trans_i$ · stage)

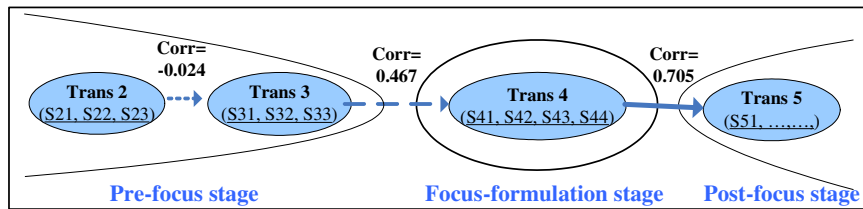


Fig. 4. Changes in task stages.

*Trans*₂, *Trans*₃, *Trans*₄ and *Trans*₅ are sampled to explain how the worker’s task-stage can be determined, as shown in Fig. 4:

- (1) *Trans*₂ and *Trans*₃: The correlation between *Trans*₂ and the first session of *Trans*₃ is -0.024 (low), which indicates that the transactions belong to very different patterns between transactions. Because the worker’s task-stage in *Trans*₂ is the pre-focus stage, we can infer that the worker’s task-stage in *Trans*₃ is also the pre-focus stage.
- (2) *Trans*₃ and *Trans*₄: The correlation between *Trans*₃ and the opening session of *Trans*₄ is 0.467 (moderate), which indicates that there are moderately similar patterns between transactions. Therefore, we can infer that the worker’s task-stage in *Trans*₄ is the task formulation stage.
- (3) *Trans*₄ and *Trans*₅: The correlation between *Trans*₄ and the first session of *Trans*₅ is 0.705 (high), which indicates that the transactions involve very similar patterns. Because the worker’s previous task-stage is the task formulation stage and the correlation between *Trans*₄ and *Trans*₅ is high, we can infer that the worker’s task-stage of *Trans*₅ is in the post-focus stage.

4.4. Knowledge support based on task-stages

In this section, we explain how the system provides document support based on the worker’s task-stage. Specifically, the system considers the worker’s task-stage and task-relevant topics of that stage to adjust the task profiles. The new task profile of the target task, denoted as \vec{S}_{p+1} , is generated by the profile adaptation equation formula defined in Eq. (5). The equation considers the worker’s task stage and the task-relevant topics of each stage. The parameter λ is used to adjust the relative importance of task-relevant topics and the temporal profile. As the task progresses, the content of the temporal profile becomes more important than that of the task-relevant topics. That is, the value of λ decreases and the influence of the temporal profile increase as the task progresses. Thus, we predefine the λ based on the changes in the task-stage. As shown below, we set λ at 0.8, in the task pre-focus stage, and at 0.5, in the task formulation stage. Finally, while a worker is in the post-focus stage, we set $\lambda = 0.2$; that is, the content of the worker’s access document is more important than that of the domain topics.

$$\begin{aligned}
 \vec{S}_{p+1} &= \alpha \vec{S}_p + \lambda \vec{R} + (1 - \lambda) \overrightarrow{Trans}_i \\
 \vec{R} &= \sum_{\forall topic_j \in Tr} w_{p+1}(topic_j) \vec{topic}_j \\
 \lambda &= \begin{cases} 0.8 & \text{when } Trans_i \cdot stage = task_pre_focus_stage \\ 0.5 & \text{when } Trans_i \cdot stage = formulation_stage \\ 0.2 & \text{when } Trans_i \cdot stage = task_post_focus_stage \end{cases}
 \end{aligned}
 \tag{5}$$

Note that, since we adopt the vector space model from the field of information retrieval (Baeza-Yates and Ribeiro-Neto, 1999), the notations with a superscript arrow denote the feature vectors of keyword terms. Each term represents a dimension in the space, i.e., the vector space model, where \vec{S}_p and \vec{S}_{p+1} denote the task profile of the target task at time p and $p + 1$, respectively. The new task profile of the target task, denoted as \vec{S}_{p+1} , is generated by the task profile before time $p + 1$, the relevant feature vector of \vec{R} and \overrightarrow{Trans}_i . The \vec{R} is derived

from the task corpora (i.e., the bag-of-words used to describe each task topic) of relevant task topics, and the temporal profile, \overline{Trans}_i , where $w_{p+1}(topic_j)$ denotes the degree of relevance of $topic_j$ to the target task. T_r represents the set of task-relevant topics. \overline{Trans}_i denotes the temporal profile (the feature vector of weighted terms) derived from the documents accessed in transaction i , as described in Section 4.2. The parameter α is set to 1, or the correlation value of task-needs patterns between time p and time $p + 1$.

The generated task profile is the system kernel that streamlines knowledge retrieval activities to further realize task-stage knowledge support. A task profile specifies the key subjects of the task at hand, and models the information needs of knowledge workers based on the task-stages. Fig. 8 in Appendix C shows the interface of knowledge support based on task-stages in our system.

5. Experiment setup

To evaluate the effectiveness of the proposed knowledge support model, we conducted an empirical investigation of its application to our problem domain. Section 5.1 reviews the experiments' objectives and procedures. We then describe the data set, provide information about the participants, and discuss the evaluation metrics.

5.1. Experimental objective and procedure

Experiments were performed to determine if the proposed *task-stage identification technique* can provide effective knowledge support. The technique was compared with the traditional relevance feedback (RF) technique used in IF systems (discussed in Sections 1 and 2). To evaluate the effectiveness of the proposed method, we follow the procedure of the standard relevance feedback (RF) technique in the IF area. The RF technique is widely adopted in IF studies to learn users dynamic interests, preferences or information needs. Generally, most studies in this area adopt the *Rocchio* method as the baseline to compare the performance of a proposed method (Salton and Buckley, 1990; Widyantoro and Yen, 2005; Yang, Yoo, Zhang, and Kisiel, 2005; Zhan, 2004). Accordingly, we use the well-known classical relevance feedback methods proposed by Rocchio (1971) and Ide (1971) for query reformulation/expansion, as listed in the (A1) and (A2) of the Appendix A. The *Linear-0* method is similar to the *Rocchio* method, except that the irrelevant feedback part of the equation is removed. The reason is that most studies suggest that information about relevant documents is more important than the content of irrelevant documents (Salton and McGill, 1983; Salton and Buckley, 1990; Yu, Luk, and Cheung, 1976).

Furthermore, we incorporate the characteristics of the long-term project/task domain into the classical relevance feedback equation to fit our problem domain. Accordingly, we designed an experiment to evaluate the effectiveness of providing knowledge support based on task-relevant topics, the worker's task-stage, and the correlation values between stages. The related parameters are incorporated into the given profile adaptation equation, i.e., Eq. (5). Consequently, the *Linear-0.5*, *Stage*, and *Stage-C* methods are compared with the baseline *Linear-0* method. Table 3 lists the four methods compared in this work: one is the baseline method and others are based on the proposed technique. Note that there are two variants in Eq. (5). One is α , which is used to incorporate the vector of the profile accumulated from the initial time to time p into the final profile \vec{S}_{p+1} of time $p + 1$, where p is the time previous to the current time $p + 1$. Based on the basic RF technique, it is effective to set α to 1 (Witten et al., 1999). However, when considering the effect of task-stages in the profile adaptation process, we set α to the *correlation value* between the transactions in the proposed technique. The second variant is λ , which is used to obtain a profile at time $p + 1$ (i.e., from time p to the current time). The profile is a vector constructed by averaging the document vector of feedback on the documents and the vector of each relevant topic's profile. Thus, if we set λ to 0, it indicates that we only consider the document vector of workers' implicit or explicit feedback on documents. This is the same as the basic RF technique. Otherwise, it will consider the proportion of relevant topics' profile based on task-stage.

We compare four methods in this work: the *Linear-0* method (i.e., the baseline method) and three others based on the proposed technique, namely, the *Linear-0.5*, *Stage*, and *Stage-Correlation* methods. The *Linear-0* (baseline) method is an incremental learning process that learns a user's current information needs from feedback about the recommended information (i.e., documents), and updates the user model for future infor-

Table 3

The methods used in each experiment (The experiments were conducted by following the procedure outlined in Section 5.1)

Technique	Method	Description	Parameter Setting
<i>Incremental learning technique</i> (baseline technique, traditional IF technique)	① Similar to standard Rocchio algorithm in relevance feedback	<ul style="list-style-type: none"> ✓ Learns a worker's information needs by considering a his/her feedback on documents ✓ The profile is a vector constructed by averaging the document vector of implicit or explicit feedback on documents, i.e. $\lambda = 0$ 	<i>Linear-0 method</i> with $\alpha = 1, \lambda = 0$ $\vec{S}_{p+1} = \alpha \vec{S}_p + \lambda \vec{R} + (1 - \lambda) \overrightarrow{Trans}_i$ (5)
	② Similar to standard Rocchio algorithm in relevance feedback	<ul style="list-style-type: none"> ✓ Learns a worker's information needs by considering his/her feedback on documents and task-relevant topics ✓ The profile is a vector constructed by averaging the document vector of feedback on documents and the vector of each relevant topic's profile, i.e., $\lambda = 0.5$ 	<i>Linear-0.5 method</i> with $\alpha = 1, \lambda = 0.5$ $\vec{S}_{p+1} = \alpha \vec{S}_p + \lambda \vec{R} + (1 - \lambda) \overrightarrow{Trans}_i$ (5)
<i>IF technique with task-stage identification</i> (our proposed technique)	③ Stage method	<ul style="list-style-type: none"> ✓ Document support considers a worker's task-stage, as described in Section 4.4 ✓ Adjusts the relative importance of the vector of feedback on documents and the vector of each relevant topics' profile based on a user's task-stage, i.e. adjusting λ based on task-stage 	<i>Stage method</i> with $\alpha = 1$, and $\lambda = 0.2, 0.5$, or 0.8 based on the task-stage of \overrightarrow{Trans}_i $\vec{S}_{p+1} = \alpha \vec{S}_p + \lambda \vec{R} + (1 - \lambda) \overrightarrow{Trans}_i$ (5)
	④ Stage-C method	<ul style="list-style-type: none"> ✓ Document support considers a worker's task-stage and the correlation value between transactions, as described in Section 4.4 ✓ Adjusts the relative importance of the vector of feedback on documents and the vector of each relevant topics' profile based on users' task-stage, i.e. adjusting λ based on task-stage ✓ Considers the correlation value by setting α to the correlation value between transactions instead of setting to 1 	<i>Stage-C method</i> with $\alpha = \text{corr}(Pat_{Trans_{i-1}}, Pat_{Trans_i}^S)$, $\lambda = 0.2, 0.5$, or 0.8 based on the task-stage of \overrightarrow{Trans}_i $\vec{S}_{p+1} = \alpha \vec{S}_p + \lambda \vec{R} + (1 - \lambda) \overrightarrow{Trans}_i$ (5)

Notes: where $\vec{R} = \sum_{\forall topic_j \in T} w_{p+1}(topic_j) \vec{topic}_j$ (as shown in (5)).

mation filtering. The method only considers a worker's feedback on documents. In contrast, the *Linear-0.5* method considers a worker's feedback on documents and task-relevant topics; that is, two kinds of information, the feature sets derived from documents and the topic profiles, are equally important in profile adaptation. Both methods update the worker's task profile using the RF technique's incremental learning process.

Learning methods based on incremental learning techniques consider a worker's feedback without considering his/her current task-stage. In this work, however, changes in a worker's task-stage and the task-relevant topics are incorporated into the traditional RF technique. Thus, the proposed system considers a worker's feedback, the task-stage, and the task-relevant topics for each stage in order to provide more effective information filtering. We designed two methods based on the *IF technique with task-stage identification*: the *Stage* method, and the *Stage-Correlation (Stage-C)* method. Technically, the worker's task-stage will influence the parameter λ in Eq. (5). The *Stage* method considers the worker's current task-stage to adjust the relative importance of task-relevant topics and his/her feedback on documents, i.e., the temporal profile discussed in Section 4.2. The *Stage-C* method, which is based on the *Stage* method, not only considers the worker's current task-stage to adjust the relative importance of task-relevant topics and his/her feedback on documents, but also incorporates the correlation value into the profile adaptation equation, i.e., Eq. (5). Specifically, the parameter α is set to the value of the correlation value between transactions, instead of being set to 1, as in the other methods.

5.2. Study design, procedure and evaluation metric

Knowledge workers usually require a substantial amount of time (e.g., a year) to accomplish knowledge-intensive work tasks. The application domain and the characteristics of the knowledge-intensive work task restrict the size of the data sample and the number of participants in the experiments.

5.2.1. Study design

The subjects were twelve graduate students in the Department of Information Management who were engaged in different tasks, such as a survey of information technology service management, text analysis for business intelligence, product recommendation, and deployment of knowledge management systems. The types of tasks include system development, thesis writing, and project surveys, which are knowledge-intensive tasks. To accomplish such tasks, knowledge workers usually require a great deal of task-relevant information. In addition, a new task in the laboratory or a project team's task generally has some degree of similarity with the previous tasks. As observed in our previous study, both experienced workers and novices in the work environment benefit from previous tasks (Liu and Wu, 2008). Thus, reusing knowledge about previous tasks (i.e., executed tasks in this work) is the key to providing effective knowledge support when workers perform new tasks.

In this work, we observed the worker's long-term document access behavior patterns in the presented system after they were assigned a new task. In our design, when the worker wants to access or share documents in the presented system, he/she needs to select one task from the current task set. Then, the system records his/her search behavior for the specific task; therefore, the system can distinguish users' search behavior for different executing task. A new task (*current task*) may be similar to some previous tasks (*executed tasks*), but the worker is not repeating the previous task. For example, if a worker is assigned a “*ITIL-based Context-aware Knowledge Recommendation*” task, he will reference previous tasks about the “concept of context-aware retrieval”, “recommendation theory”, or “issues in the ITIL”. We do not instruct users to do anything; that is, the system traces and records their search behavior and then delivers documents based on their feedback behavior, as described in Section 4. The system periodically generates a task profile (i.e., a set of weighted keywords) based on users' feedback behavior and provides knowledge support (i.e., document recommendations) based on their profiles and task-stage.

5.2.2. Study procedure

Since executing a task takes a long time, we chose the subjects according to the task-stage they were in, i.e., the pre-focus, focus formulation, or post-focus task stages. To evaluate the effectiveness of the task-stage knowledge support model, we selected four evaluation subjects who were in the same stage, but performing different *current tasks*. Specifically, we selected four subjects who were in the pre-focus stage, four subjects

Table 4
Descriptions of the evaluations tasks

Evaluations	Workers' task-stage	Start date	End date (evaluation time)	Task types
Case 1	Pre-focus	2004/10	2005/03	Thesis writing
Case 2	Pre-focus	2003/09	2003/12	Thesis writing
Case 3	Pre-focus	2004/11	2005/03	Thesis writing
Case 4	Pre-focus	2002/12	2003/09	Project survey
Case 5	Focus-formulation	2004/10	2005/03	System development
Case 6	Focus-formulation	2003/09	2004/04	Thesis writing
Case 7	Focus-formulation	2003/10	2005/03	Thesis writing
Case 8	Focus-formulation	2003/06	2004/03	Project survey
Case 9	Post-focus	2003/09	2005/12	Thesis writing
Case 10	Post-focus	2003/09	2005/04	Thesis writing
Case 11	Post-focus	2003/09	2005/10	System development
Case 12	Post-focus	2003/06	2004/09	Project survey

who were in the focus formulation stage, and four subjects who were in the post-focus stage. Accordingly, there are 12 subjects in this experiment, i.e., we conducted four evaluations for each task-stage. Table 4 details the subjects' information, including the task-stages they were in, and the length of the evaluation period. Note that subjects may stay a long time in a stage if they encounter problem with the task-at-hand. The effectiveness of knowledge support is measured in terms of the precision and recall rates, as in information retrieval research (Rijsbergen, 1979; Riloff and Lehnert, 1994). For each evaluating task, the precision rate is the percentage of retrieved items (tasks or documents) that are relevant compared to the total number of retrieved documents. The recall rate is the percentage of the total number of relevant documents retrieved to the total number of relevant documents, specified by experts.

$$\text{precision}_{e_r} = \frac{|\text{retrieved documents that are relevant}|}{|\text{total retrieved documents}|} \quad (6)$$

$$\text{recall}_{e_r} = \frac{|\text{retrieved documents that are relevant}|}{|\text{total relevant documents specified by experts}|} \quad (7)$$

For each evaluation task, we asked domain experts and experienced workers to manually label the documents as relevant or irrelevant to the task. This is a time consuming task, but it is a good way to ensure the quality of our answer set for evaluating each task. Finally, on average, 40–60 documents were labeled as relevant by the domain experts and experienced workers for each evaluation task. The experiment results summarize the observations based on task-stages and various levels of top- N document support.

5.2.3. Data and participants

Task-relevant codified knowledge consists of documents in an organization's knowledge repository that have been accessed via the executed task set, i.e., historical tasks. In this work, the tasks are related to writing research papers or conducting research projects, so we selected evaluation subjects who were engaged in a current task. A task denotes either an *executed task* or a *current task*. An *executed task* is a historical *task* already accomplished within the organization, whereas a *current task* is a *task* at hand. More details are given in the Liu and Wu (2008). We selected approximately 600 documents from the 35 executed task sets in our laboratory for the period 2002–2006. Each document contained an average of 90 distinct terms after information extraction, and document pre-processing (i.e., case folding, stemming, and stop word removal). Note that a large number of documents are accessed and generated during a task's execution. After completing a task, the task-related experts and knowledge workers identify key documents. As a result, the task repository only preserves around 15–20 key documents, which are selected from the set of documents for each *task*.

6. Experiment results and implications

We conducted the experiment to evaluate the effectiveness of the proposed *IF technique with task-stage identification* using the four methods described in Section 5.1. As described in Section 3, changes in a worker's task-stages are determined by the proposed correlation of task-needs pattern analysis. The result of stage identification influences the parameters α and λ in profile adaptation, as shown in Table 3.

6.1. Experiment results

In this experiment, we compare the *Linear-0*, *Linear-0.5*, *Stage*, and *Stage-C* methods. The *Linear-0* is based on the traditional *incremental learning technique*. The *Linear-0.5* method is similar to traditional *incremental learning technique*; it also considers a worker's feedback on task-relevant topics. The *Stage* and *Stage-C* Methods is based on the *IF technique with task-stage identification*. Moreover, for the *Stage-C* method, the correlation value of the worker's task-need patterns across transactions is incorporated into the technique.

6.1.1. Document support: observations based on task-stages

Fig. 5 shows the performance of the three methods in terms of precision, and recall. In addition, Fig. 6 shows the result of knowledge support for document-retrieval based on the three task stages. The top-30 retrieved documents are used to evaluate the performance of each method:

Observation 1: Fig. 5 shows that the average values of the precision and recall of the *Stage-C* method are better than those of the other methods, especially the *Linear-0* and *Linear-0.5* methods. Meanwhile, the *Linear-0* method has better performance than the *Linear-0.5* method.

Observation 2: Taking a further look of the experiment, we evaluate the performance of the four methods for each task-stage. Fig. 6 shows that the *Stage-C* method performs better than the other three methods over three task-stages. Interestingly, the *Stage* method only has far better performance than the *Linear-0* and *Linear-0.5* methods in the pre-focus stage. Furthermore, the *Linear-0* method even has better performance in the task post-focus stage. The result shows that (1) Overall, the performance of the *Stage-C* method for each task-stage is better than that of the other two methods, and (2) The incremental learning technique, i.e., the *Linear-0* method, has worse performance in the early stage of the task execution, whereas has better performance in the task post-focus stage, i.e., the worker has dedicated on the specific research topic/issue.

Discussions: The experimental result indicates that a worker's topic requirements may vary in the early stage of a task's execution, i.e., we observed that a worker in the first task-stage may have many task-needs in relation to topics and may change topics. Thus, the correlation value, $\alpha = \text{corr}_u(\text{Pat}_{\text{Trans}_p}, \text{Pat}_{\text{Trans}_{p+1}}^S)$, between transactions is negative, which indicates that the worker has completely different task-needs patterns in relation to topics in the first stage. Obviously, the effect of changing topics cannot be reflected by the *incremental learning techniques*. Therefore, the *traditional incremental learning technique*, i.e., adjusting the task profile based the worker's access documents, is not suitable for learning the worker's task-needs in the early stage of task execution. But the correlation value between transactions can be reflected in the *Stage-C* method. That is, if the correlation value between transactions is negative, the feature set in the task profile of the previous transaction is subtracted from the feature set in the task profile of current transaction to remove irrelevant feature terms from the profile. Thus, the result confirms that it is reasonable to incorporate the correlation value into the profile adaptation process. Furthermore, the incremental learning technique, i.e., the *Linear-0* method, performs better than the *Linear-0.5* and *Stage* methods in the post-focus stage. This suggests that once workers have focused on specific topics, learning their task-

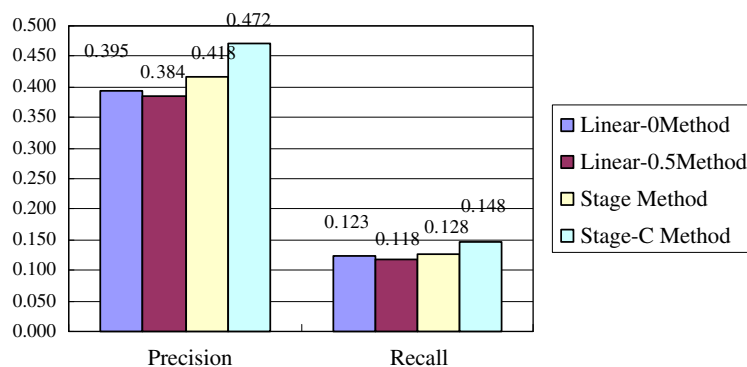


Fig. 5. Document support based on the average performance value of each method (the y axis shows the performance value).

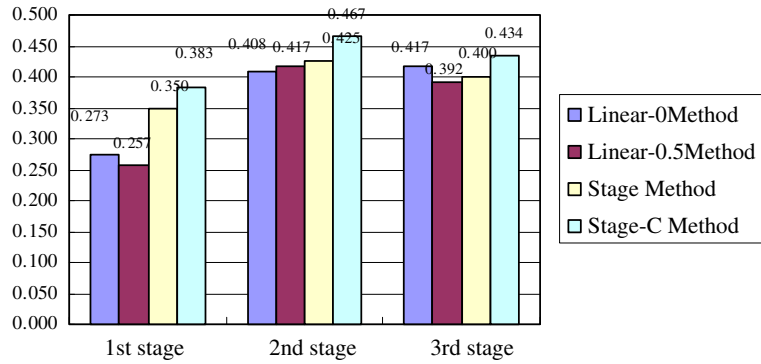


Fig. 6. Document support based on the average performance value of each stage (the y axis shows the precision).

needs from their feedback on documents is reliable. Therefore, the equation favors the weight of the accessed document set, \overrightarrow{Trans}_i by decreasing the value of parameter λ in Eq. (5).

6.1.2. Document support: observations on various levels of top-N support

Fig. 7 shows the precision of the three methods obtained by averaging the three task stages for various levels of top-N retrieved document support. In addition, Table 5 shows the results of task-relevant document support based on the three stages for various levels of top-N retrieved document support:

Observations 1: Fig. 7 shows the average precision values of each evaluation case in each stage under various levels of document support, i.e., top-5, 10, 20 and 30 document support. The Stage-C method outperforms the other three methods, especially for top-5 document support. Thus, it provides more effective knowledge support than the other methods. Moreover, it also shows that the worker could get task-relevant documents without browsing lots of retrieved documents by the Stage-C method.

Observations 2: Table 5 shows that the average values of precision and recall of the Stage-C method under various levels of top-N, i.e., top-5, 10, 20 and 30, exceeds those of the other methods. Thus, the Stage-C method provides more effective knowledge support than the other methods under various levels of top-N document support in each task-stage. Interestingly, the Linear-0 method performs better than the Linear-0.5 and Stage methods in the task post-focus

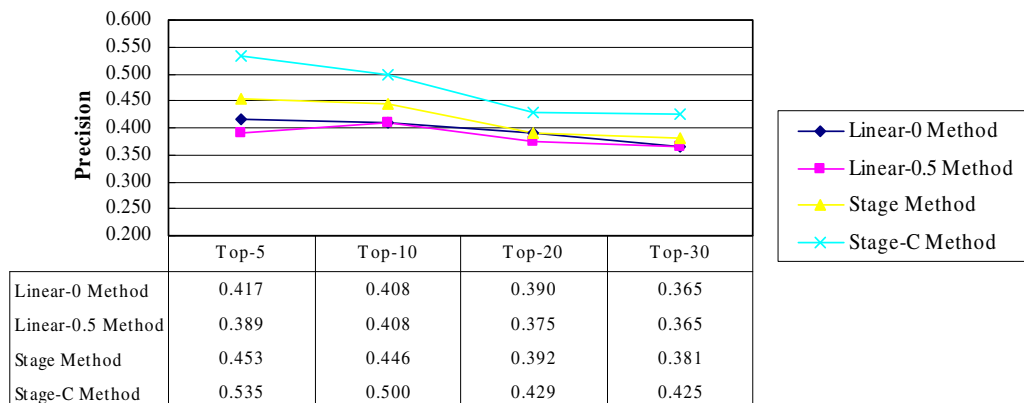


Fig. 7. The average performance value under various levels of top-N support.

Table 5
Document support based on the average performance value under various levels of top- N support in each task-stage (experimental one)

Stage	Top- N	Linear-0 method		Linear-0.5 method		Stage method		Stage-C method	
		Pre.	Re.	Pre.	Re.	Pre.	Re.	Pre.	Re.
1st stage (pre-focus)	Top-5	0.450	0.054	0.350	0.042	0.467	0.096	0.442	0.089
	Top-10	0.375	0.093	0.375	0.090	0.350	0.094	0.425	0.115
	Top-20	0.288	0.141	0.313	0.149	0.300	0.130	0.363	0.178
	Top-30	0.270	0.200	0.278	0.204	0.308	0.195	0.375	0.214
2nd stage (focus formulation)	Top-5	0.350	0.026	0.525	0.044	0.550	0.051	0.600	0.058
	Top-10	0.450	0.074	0.500	0.095	0.525	0.095	0.525	0.095
	Top-20	0.446	0.184	0.450	0.163	0.450	0.138	0.475	0.178
	Top-30	0.408	0.226	0.417	0.229	0.425	0.242	0.467	0.279
3rd stage (post-focus)	Top-5	0.450	0.025	0.292	0.090	0.342	0.094	0.563	0.045
	Top-10	0.400	0.074	0.350	0.092	0.463	0.114	0.550	0.103
	Top-20	0.438	0.157	0.363	0.076	0.425	0.133	0.450	0.185
	Top-30	0.417	0.222	0.400	0.148	0.409	0.154	0.434	0.236
Total average		0.395	0.123	0.384 (-2.78%)	0.118	0.418 (+5.82%)	0.128	0.472 (+19.49%)	0.148

stage in general. This result is in accordance with Observation 2 in Section 6.1.1. Since the worker has dedicated on specific research topics, it is effective to provide task-relevant documents based on his/her feedback behaviors on documents.

6.2. Implications

Overall, the experiment results demonstrate that the proposed long-term document support model for task-stages is effective. Since the *Stage-C* method achieves the best performance in each evaluation case, the proposed *task-stage identification technique* can provide more task-relevant documents than the *incremental learning technique*. In addition, the smaller the number of support documents, the higher the precision rate of each method will be; that is, the system has a better retrieval capability with fewer supporting documents. On the other hand, the fewer number of support documents, the lower recall rate for each method under various task stages. The reason is we asked domain experts and experienced workers to manually label documents as relevant or irrelevant to the task. On average 40–60 were labeled as relevant for each evaluation task, as mentioned in Section 5.2. One of the evaluation cases even with 30 relevant documents which are labeled by the experts. Therefore, it is not easy to achieve high recall value. Due to the higher precision rate performance based on the experimental result, the system is suitable for adopting in a work environment in which workers are under pressure to find task-relevant documents and do not have time to review a large number of retrieved documents. Moreover, with the designed relevance feedback technique in the presented system, the problem of low recall rate could be resolved by adjust the worker's profile based on his/her explicit and implicit feedback behaviors. Furthermore, the results are in accordance with our preliminary assumptions that we could build an interactive knowledge support system to retrieve task-relevant documents based on the proposed techniques to perform more precise information filtering. In the future, we would like to conduct on-line evaluations to explore the issues of interactive information seeking behavior according to the proposed ideas.

7. Conclusion and future work

Information systems that rely on traditional IR technology and user models (e.g., user profiles) to support workers' long-term information needs raise some fundamental and challenging issues. The most important issue is that it is difficult to model a worker's long-term information needs based on his/her feedback

without considering the working context. To provide more effective long-term document support, we incorporated a task-stage identification technique into the document support process in each task-stage. Based on Vakkari's study (2000), we observed the changes in laboratory workers' task-stages over a 1-year period. Using a task-needs pattern similarity analysis method, we identified three stages, namely, the pre-focus, focus formulation, and post-focus task-stages. The experiment results show that in the pre-focus task-stage, there is a low or negative correlation between search sessions and transactions, whereas there is at least a moderate correlation between these two factors in the task formulation and post-focus stages. Based on this finding, we have proposed a task-stage knowledge support model that adjusts a worker's task profile according to task-relevant topics and the specific task-stage. Preliminary experiments were conducted to compare methods that deliver task-relevant documents based on task-stages and the traditional information filtering (IF) incremental learning technique. The results confirm that incorporating a worker's task-stages into the traditional IF incremental learning technique can effectively support the worker's long-term information needs.

A number of issues still need to be addressed. First, context-based knowledge support that considers the context of activities, roles, and work-related skills when providing context-aware knowledge access and retrieval is an important issue that requires further research. Contemporary KMS increasingly emphasize the organization of all possible task-specific knowledge by supporting context-aware knowledge access and retrieval (Ahn et al., 2005; Alvarado et al., 2004; Kwan and Balasubramanian, 2003). Future studies could extend the proposed approach to support context-aware or process-aware delivery of task-relevant knowledge. Second, more task-stage features should be considered to refine the adaptation of the task profile during the task's performance. As Vakkari's studies show, information search terms and relevance judgments vary according to the task-stage. Some of the above features may be crucial for enhancing the task-stage knowledge support model; therefore, we will conduct further investigations by considering the features of each stage. Moreover, reusing knowledge assets extracted from historical tasks is important in providing effective knowledge support for tasks. In our ongoing work, we are developing a mining technique for discovering task-stage needs from historical tasks. The proposed method uses information retrieval techniques and a modified hierarchical agglomerative clustering algorithm to identify task-stage needs by analyzing codified knowledge (documents) accessed or generated during a task's execution (Liu, Wu, and Chen, 2006). We are also developing a method to identify variations in topic-needs across task-stages based on the Vakkari (2000) and Vakkari et al. (2003) information seeking theory. The objective is to determine the influence of broad and specific topics in different task-stages.

Finally, we should acknowledge the limitations of this study. It was an exploratory longitudinal study, and we implement them in real-world settings, i.e., a laboratory in which multiple projects were executed by different workers. As a result, it was difficult to choose proper tasks and subjects as our investigation targets. Using filtering rules devised by experts, a few tasks and researchers were chosen as test subjects. Although the sample size in this type of research is limited (Hansen and Järvelin, 2005; Jansen, 2005; Vakkari et al., 2003), the experiment results motivate us to conduct further research on long-term document support in different task-stages. In addition, with regard to the selection of tasks (i.e., evaluation cases), we need to emphasize that the tasks should be in the same research domain. They cannot relate to topics outside the task-based domain ontology because we focus on the effective reuse of knowledge about previous tasks (i.e., historical tasks in this work) to support the execution of new tasks. In our future work, we will consider knowledge support for innovation tasks, as it is beyond the scope of this study. Moreover, Kelly and Fu's (2007) relevance feedback study shows that eliciting a user's information needs based on his/her feedback can improve retrieval performance significantly. We will refine the experiment procedure to further evaluate the effectiveness of the proposed techniques for real-time and interactive document delivery. Ideally, we would like to conduct on-line evaluations to explore issues related to interactive information seeking based on the proposed task-stage technique and identify workers' task-needs with the aid of an interactive process for exploring specific topics.

Acknowledgements

This research was supported by the National Science Council of Taiwan government under the Grant NSC 95-2416-H-009-002 & NSC 95-2416-H-030-012.

Appendix A. Relevance feedback techniques

Relevance feedback improves search effectiveness through query reformulation (Rocchio, 1971). Various studies have demonstrated that relevance feedback applied in the vector model is an effective technique for information retrieval (Rocchio, 1971; Salton and Buckley, 1990). Eqs. (A1) and (A2) illustrate two classical relevance feedback methods designed by Rocchio (1971) and Ide (1971), respectively. A modified query vector \vec{q}_m is derived using the relevance of documents (as feedback) to adjust the query vector \vec{q} (Baeza-Yates and Ribeiro-Neto, 1999):

$$\text{Standard_Rocchio: } \vec{q}_m = \alpha\vec{q} + \beta \frac{1}{|D_r|} \sum_{\forall d_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_n|} \sum_{\forall d_j \in D_n} \vec{d}_j \quad (\text{A1})$$

$$\text{Ide_Dec_Hi: } \vec{q}_m = \alpha\vec{q} + \beta \sum_{\forall d_j \in D_r} \vec{d}_j - \gamma \max_{\text{irrelevant}} (\vec{d}_j) \quad (\text{A2})$$

where D_r denotes the set of relevant documents and D_n represents the set of irrelevant documents according to user judgment. $|D_r|$ and $|D_n|$ represent the number of documents in the sets D_r and D_n . Meanwhile, α , β , γ are tuning constants. The function of $\max_{\text{irrelevant}}$ returns the most irrelevant document. The two methods produce similar results (Baeza-Yates and Ribeiro-Neto, 1999).

Appendix B. Task topics in the research domain

Herein, we listed the 35 task topics in this research domain

Topic ID	Topics (previously executed tasks)
T000000001	Role-based intranet information service
T000000002	Automatic classification for video data
T000000003	Agent-based internet search for goods information
T000000004	A study on internet healthcare commerce: integration of healthcare EDI and SET
T000000005	A flexible architecture for intermediary based electronic commerce
T000000006	The development of task-oriented intranet information service systems integrating with information warehouse
T000000007	Workflow modeling based on XML and rules
T000000008	A case study of introducing ERP systems in enterprises
T000000009	Research on personalized merchandise information filtering in electronic catalog systems
T000000010	Design and implementation of XML based electronic catalogs
T000000011	Designing authorization rules to achieve separation of duty in task-based access control models
T000000012	An implementation of authorization management in task-based access control models
T000000013	An implementation of multidimensional transaction analysis systems for e-brokers
T000000014	Mining association rules from three-level e-catalogs
T000000015	Deploying a data warehouse system for computer integrated manufacturing
T000000016	Mining association rules for information recommendation in enterprises
T000000017	Integrating data warehousing and data mining for web logs analysis
T000000018	Designing composite e-service platform with recommendation capability
T000000019	Multi-criteria task assignment in workflow management systems
T000000020	Coordinating inter-organizational workflows based on process-views
T000000021	Workflow modeling for virtual processes: an order-preserving process-view approach
T000000022	Towards a framework for discovering project-based knowledge maps
T000000023	A integrated framework for recommending composite e-services based on mining and collaborative filtering

(continued on next page)

Appendix B (continued)

Topic ID	Topics (previously executed tasks)
T0000000024	A study of topic maps for composite e-services
T0000000025	A implementation and comparison of collaborative filtering for recommendations
T0000000026	A study of applying feature-weighting clustering to recommender system
T0000000027	Event detection and tracking based on event hierarchy
T0000000029	Mining the changes of news events for supporting decision making
T0000000030	Discovery and support of problem-solving knowledge in e-business
T0000000031	A collaborative relevance feedback approach to task-driven recommendation
T0000000038	Applying second-degree neighborhood to alleviate the sparsity problem in collaborative filtering
T0000000043	Applying topic maps and data mining to deploy composite e-service platform
T0000000047	Implementation of task-based knowledge support system
T0000000048	Implementation of personalized recommendations for composite e-services
T0000000050	Business-to-business workflow interoperation based on process-views

Appendix C. K-Support revised interface

See Fig. 8.

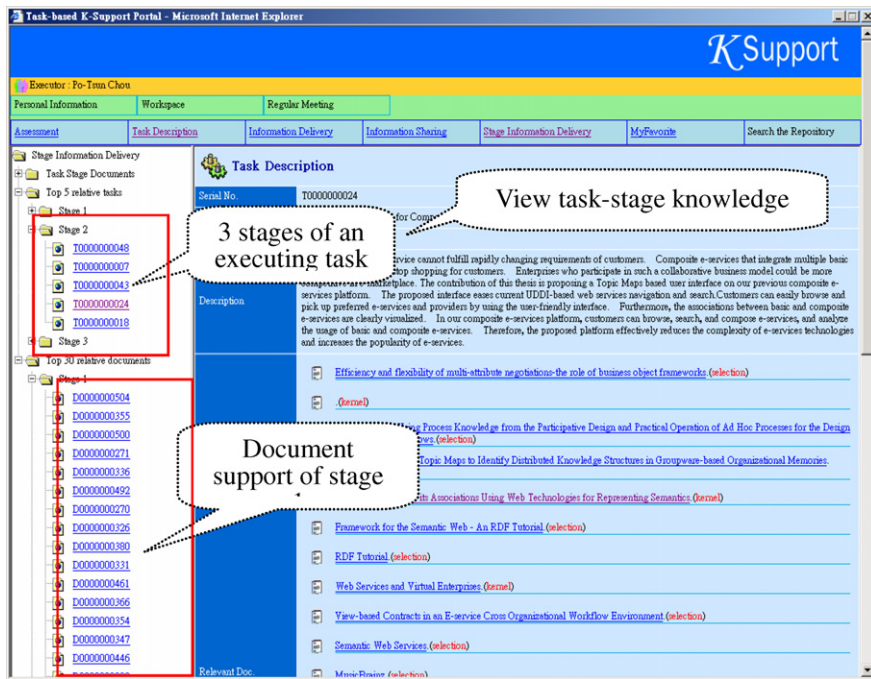


Fig. 8. Interface of knowledge support based on stages.

References

- Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., & Sintek, M. (2000). Context-aware, proactive delivery of task-specific knowledge: The KnowMore project. *International Journal on Information Systems Frontiers (ISF)*, 2(3/4), 139–162.
- Ahn, H. J., Lee, H. J., Cho, K., & Park, S. J. (2005). Utilizing knowledge context in virtual collaborative work. *Decision Support Systems*, 39, 563–582.
- Alvarado, M., Romero-Salcedo, M., & Sheremetov, L. (2004). A corporate memory based on the user profiles. *Expert Systems with Applications*, 26(1), 87–94.
- Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. New York: The ACM Press.
- Balabanovi'c, M. (1997). An adaptive Web page recommendation service. In *Proceedings of the conference on autonomous agents* (pp. 378–385).
- Belkin, N. J., & Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin?. *Communications of the ACM* 35(12), 29–38.
- Billsus, D., & Pazzani, M. (1997). User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2/3), 147–180.
- Bolloju, N., Khalifa, M., & Turban, E. (2002). Integrating knowledge management into enterprise environments for the next generation decision support. *Decision Support Systems*, 33(22), 163–176.
- Byström, K., & Järvelin, K. (1995). Task complexity affects information seeking and use. *Information Processing and Management*, 31(2), 191–213.
- Celentano, A., Fugini, M. G., & Pozzi, S. (1995). Knowledge-based document retrieval in office environment: The Kabiria system. *ACM Transactions on Information Systems*, 13(3), 237–268.
- Davenport, T. H., & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Boston, MA: Harvard Business School Press.
- De Bra, P., Houben G. J., & Dignum, F. (1997). Task-based information filtering: Providing information that is right for the job. In *Proceedings of conferentie informatie wetenschappen*.
- Fenstermacher, Kurt D. (2002). Process-aware knowledge retrieval. In *Proceedings of the 35th Hawaii international conference on system sciences (HICSS 2002)* (pp. 209–218).
- Gray, P. H. (2001). Problem-solving perspective on knowledge management practices. *Decision Support Systems*, 31(1), 87–102.
- Hanani, U., Shapira, B., & Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction (UMUAI)*, 11(3), 203–259.
- Hansen, P., & Järvelin, K. (2005). Collaborative information retrieval in an information-intensive domain. *Information Processing and Management*, 41, 1101–1119.
- Hoashi, K., Matsumoto, K., Inoue, N., & Hashimoto, K. (2000). Document filtering method using non-relevant information profile. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2000)* (pp. 176–183).
- Holz, H., Maus, H., Bernardi, A., & Rostanin, O. (2005). A lightweight approach for proactive, task-specific information delivery. In *Proceedings of the fifth international conference on knowledge management (I-Know 2005)*.
- Ide, E. (1971). New experiments in relevance feedback. In G. Salton (Ed.), *The SMART retrieval system: Experiments in automatic document processing* (pp. 337–354). Englewood Cliffs, NJ: Prentice Hall.
- Ingwersen, P., & Järvelin, K. (2005). *The turn: Integration of information seeking and retrieval context*. Dordrecht, The Netherlands: Springer.
- Jansen, B. J. (2005). Seeking and implementing automated assistance during the search process. *Information Process and Management*, 41(4), 909–928.
- Kelly, D. (2004). *Understanding implicit feedback and document preference: A naturalistic user study*. PhD dissertation, Rutgers University.
- Kelly, D., & Fu, X. (2007). Eliciting better information need descriptions from users of information systems. *Information Processing and Management*, 43(1), 30–46.
- Kuhlthau, C. (1993). *Seeking meaning*. Noorwood, NJ: Ablex.
- Kwan, M. M., & Balasubramanian, P. (2003). KnowledgeScope: Managing knowledge in context. *Decision Support Systems*, 35(4), 467–486.
- Liu, D.-R., & Wu, I.-C. (2008). Collaborative relevance assessment for task-based knowledge support. *Decision Support Systems*, 44(2), 524–543.
- Liu, D.-R., Wu, I.-C., & Chen, W.-H. (2006). *Mining and supporting task-stage knowledge: A hierarchical clustering technique. Lecture notes in artificial intelligence: Practical aspect of knowledge management* (vol. 4333). Berlin, Heidelberg: Springer-Verlag, pp. 178–188.
- Liu, D.-R., Wu, I.-C., & Yang, K.-S. (2005). Task-based K-support system: Disseminating and sharing task-relevant knowledge. *Expert Systems with Applications*, 29(2), 408–423.
- Markus, M. L. (2001). Toward a theory of knowledge reuse: Types of knowledge reuse situation and factors in reuse success. *Journal of Management Information Systems*, 18(1), 57–94.
- Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender system. *ACM Transactions on Information Systems*, 22(1), 54–88.
- Mostafa, J., Mukhopadhyay, S., Lam, W., & Palakal, M. (1997). A multi-level approach to intelligent information filtering: Model, system and evaluation. *ACM Transactions on Information Systems*, 15(4), 368–399.
- Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1), 14–37.
- Park, J., & Hunting, S. (2003). *XML topic maps: Creating and using topic maps for the Web*. Boston, MA: Addison-Wesley.

- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130–137.
- Rijsbergen, C. J. V. (1979). *Information retrieval*. London: Butterworths.
- Riloff, E., & Lehnert, W. (1994). Information extraction as a basis for high precision text classification. *ACM Transaction on Information System*, 12(3), 296–333.
- Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART retrieval system: Experiments in automatic document processing* (pp. 313–323). Englewood Cliffs, NJ: Prentice Hall.
- Ruthven, I., Lalmas, M., & van Rijsbergen, C. J. (2003). Incorporating user search behavior into relevance feedback. *Journal of the American Society for Information Science and Technology*, 54(6), 529–549.
- Salton, G., & Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5), 513–523.
- Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4), 288–297.
- Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGrawHill Book Co.
- Spies, M., Clayton, A. J., & Noormohammadian, M. (2005). Knowledge management in a decentralized global financial services provider: A case study with Allianz Group. *Knowledge Management Research and Practice*, 3(1), 24–36.
- Vakkari, P. (2000). Relevance and contribution information types of searched documents in task performance. In *Proceedings of the 23rd annual international ACM SIGIR conference (SIGIR 2000)* (pp. 2–9). Athens, New York, NY: ACM.
- Vakkari, P., Pennanen, M., & Serola, S. (2003). Changes of search terms and tactics while writing a research proposal: A longitudinal case study. *Information Processing and Management*, 39(3), 445–463.
- Wang, P. L., & Soergel, D. (1998). A cognitive model of document use during a research project. Study I. Document selection. *Journal of the American Society for Information Science and Technology*, 49(2), 115–133.
- Wang, P. L., & White, M. D. (1999). A cognitive model of document use during a research project. Study II. Document at reading and citing stage. *Journal of the American Society for Information Science and Technology*, 50(20), 98–114.
- White, R. W., & Kelly, D. (2006). A study on the effects of personalization and task information on implicit feedback performance. In *Proceedings of the 15th ACM international conference on information and knowledge management (CIKM 2006)* (pp. 297–306).
- White, R. W., Jose, J. M., & Ruthven, I. (2003). Adapting to evolving needs: Evaluating a behavior-based search interface. In *Proceedings of the 17th annual conference on human computer interaction (HCI 2003)* (pp. 125–128).
- White, R. W., Kules, B., Drucker, S. M., & Schraefel, M. C. (2006). Supporting exploratory search. *A Special Section of the Communications of the ACM*, 49(4), 36–39.
- Widyantoro, D. H., & Yen, J. (2005). Relevant data expansion for learning concept drift from sparsely labeled data. *IEEE Transactions on Knowledge and Data Engineering*, 17(3), 401–412.
- Widyantoro, H., Ioerger, T. R., & Yen, J. (2001). Learning user interest dynamics with a three-descriptor representation. *Journal of the American Society for Information Science*, 52(3), 212–225.
- Witten, I. H., Moffat, A., & Bell, T. C. (1999). *Managing gigabytes: Compressing and indexing documents and images* (2nd ed.). Los Alto, USA: Morgan Kaufmann Publishers.
- Yang, Y., Yoo, S., Zhang, J., & Kisiel, B. (2005). Robustness of adaptive filtering methods In a cross-benchmark evaluation. *Proceedings of the 28rd annual international ACM SIGIR conference (SIGIR 2005)* (pp.98-105). Salvador, Brazil.
- Ye, Y., & Fischer, G. (2002). Supporting reuse by delivering task-relevant and personalized information. In *Proceedings of the 24th international conference on software engineering* (pp. 513–523).
- Yu, C. T., Luk, W. S., & Cheung, T. Y. (1976). A statistical model for relevance feedback in information retrieval. *Journal of the ACM*, 23, 273–286.
- Zack, M. H. (1999). Managing codified knowledge. *Sloan Management Review*, 40(4), 45–58.
- Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning, parts 1, 2, and 3. *Information Sciences*, 8(2), 199–249; 8(3), 301–357; 9(1), 43–80.
- Zhan, Y. (2004). Using Bayesian priors to combine classifiers for adaptive filtering. In *Proceedings of the 27rd annual international ACM SIGIR conference (SIGIR 2004)* (pp. 345–352).