國 立 交 通 大 學

資訊管理研究所

博 士 論 文

以工作觀為基礎之知識支援模式與系統:
工作相關知識遞送與分享

Task-based $\mathcal{K}$-Support Model and System: Delivering and Sharing Task-relevant Knowledge

研 究 生： 吳 怡 瑾

指導教授： 劉 敦 仁

中 華 民 國 九 十 五 年 一 月

國 立 交 通 大 學

資訊管理研究所

博 士 論 文

以工作觀為基礎之知識支援模式與系統:
工作相關知識遞送與分享

Task-based *K*-Support Model and System: Delivering and Sharing Task-relevant Knowledge

研 究 生：吳怡瑾

研究指導委員會：陳彥良 博士
魏志平 博士
羅濟群 博士
楊 千 博士

指導教授：劉敦仁 博士

中 華 民 國 九 十 五 年 一 月

以工作觀為基礎之知識支援模式與系統:
工作相關知識遞送與分享

Task-based *K*-Support Model and System: Delivering and Sharing Task-relevant Knowledge

研 究 生：吳怡瑾　　Student：I-Chin Wu

指導教授：劉敦仁　　Advisor：Dr. Duen-Ren Liu

國 立 交 通 大 學
資 訊 管 理 研 究 所
博 士 論 文

A Dissertation
Submitted to Institute of Information Management
College of Management
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Information Management

January 2006

Taipei, Taiwan, Republic of China

中華民國九十五年一月

博碩士論文授權書

(國科會科學技術資料中心版本 92.2.17)

本授權書所授權之論文為本人在 國立交通大學 大學(學院) 資訊管理 系所 九十四 學年度第 一 學期取得 博 士學位之論文。

論文名稱：以工作觀為基礎之知識支援模式與系統：工作相關知識遞送與分享

1. □同意 ☑不同意 （政府機關重製上網）

   本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中心、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

   本論文為本人向經濟部智慧財產局申請專利（未申請者本條款請不予理會）的附件之一，申請文號為：____________，註明文號者請將全文資料延後半年再公開。

2. ☑同意 □不同意 （圖書館影印）

   本人具有著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鉤選，本人同意視同授權。

指導教授姓名： 劉敦仁

研究生簽名： 學號：8934502

（親筆正楷） （務必填寫）

日期：民國九十五年一月十二日

---

1. 本授權書（得自 http://sticnet.stic.gov.tw/sticweb/html/theses/authorize.html 下載或至 http://www.stic.gov.tw 首頁右下方下載）請以黑筆撰寫並影印裝訂於書名頁之次頁。
2. 授權第一項者，請確認學校是否代收，若無者，請個別再寄論文一本至台北市(106-36)和平東路二段 106 號 1702 室 國科會科學技術資料中心 王淑貞。(本授權書諮詢電話:02- 27377746)
3. 本授權書於民國 85 年 4 月 10 日送請內政部著作權委員會(現為經濟部智慧財產局)修正定稿，89.11.21 部份修正。
4. 本案依據教育部國家圖書館 85.4.19 台(85)圖編字第 712 號函辦理。

Task-based $\mathcal{K}$-Support Model and System: Delivering and Sharing Task-relevant Knowledge

Student: I-Chin Wu **Advisor:** Dr. Duen-Ren Liu

Institute of Information Management
National Chiao Tung University

ABSTRACT

In task-based business environments, a pertinent issue in deploying knowledge management system (KMS) is providing task-relevant information (codified knowledge) to fulfill the information needs of knowledge workers. Historical codified knowledge, i.e. experiences and know-how extracted from previous task executions, provides valuable knowledge for knowledge workers to accomplish tasks successfully. Accordingly, a repository of structured and explicit knowledge, especially in document form, is a widely adopted codification-based strategy for managing knowledge in KMS.

This work first discusses the issue of managing codified knowledge by building the task-oriented repository from the perspective of business task. To organize and manage task-relevant information, the repository is constructed with support from domain ontology (topic taxonomy) to effectively utilize codified knowledge. Thus, providing effective knowledge retrieval function to mitigate the difficulty of accessing knowledge items from the knowledge repository is a challenging work. Accordingly, a task-based knowledge support model is proposed to tackle the problem. The proposed model proactively delivers task-relevant codified knowledge and promotes knowledge sharing among knowledge workers in task-based business environments.

A novel **task-relevance assessment approach** *is proposed to identify the knowledge worker's information needs on tasks, for brevity, task-needs. The proposed approach generates task profiles via the collaboration of knowledge workers to analyze the relevance of tasks and codified knowledge. The approach can* *alleviate the problem of accessing needed knowledge items from vast amounts of codified knowledge. Moreover, an* **adaptive task-based profiling approach** *and a* **task peer-group analytical method** *are proposed to track workers' dynamic task-needs and identify workers' task-based peer-groups p. Knowledge workers can obtain task-relevant knowledge with the aid of task-based profiles and peer-groups. Furthermore, we seek to extend and refine our model to resolve long-term knowledge support problem. According to our empirical investigation, knowledge workers engaged in knowledge intensive task usually have different information needs during the long-term task performance. That is, another challenge of deploying KMS is to support task-relevant knowledge based on workers' task-needs at different task progress, i.e., stages or milestones. Accordingly, we proposed a task-stage knowledge support model that incorporates the information-filtering model with the identification of worker's task-stage. A* **correlation analysis method** *is proposed to identify a worker's task-stage, and an* **ontology-based topic discovery method** *is proposed to determine a worker's task-needs for specific topics of stage. Consequently, the system can be tailored to support long-term task performance.*

A task-based $\mathcal{K}$-Support portal is developed to facilitate knowledge reuse and further to streamline task execution. The portal is grounded in a research institute to support the execution of knowledge-intensive task by stimulating the operation of knowledge delivering and sharing. Moreover, various experiments have been conducted to evaluate the proposed model. The experimental results reveal that the proposed model and system can provides knowledge support in task-based environments effectively.

Keywords: Knowledge management system, Task-relevant knowledge, Codified knowledge, Task-relevance assessment, Adaptive task profile, Knowledge delivery, Knowledge sharing, Task-stage, $\mathcal{K}$-Support portal

以工作觀為基礎之知識支援模式與系統:
工作相關知識遞送與分享

研究生：吳怡瑾　　　　　　　　　　　　指導教授：劉敦仁 博士

國立交通大學資訊管理所

摘要

建構知識管理系統已是企業組織有效管理企業知識，獲取產業競爭優勢的重要策略。而企業主要是以工作為基礎來進行企業活動之運作與管理，組織人員執行各項工作以達成企業之營運目標。在以工作為基礎之企業環境，考量組織工作特性，設計適合的知識推薦機制，以提供組織人員工作相關之知識物件與資訊，是建構知識管理系統之重要議題。

一般而言，在各類知識物件中，文件為將知識外顯化的重要方式之一；此外，文件除提供豐富之資訊並且也是增加速度最為可觀之知識物件。因此，企業若能將各式知識物件以結構化方式存放至知識庫並使之外顯化，勢必能有效保存與提供組織知識資產。本研究主要設計以工作為基礎的主題分類架構(task domain ontology)…，並引入模糊分類方法，將企業內的各項知識物件與工作，配合該主題分類架構加以分類與整理。此外，為支援知識工作者克服執行工作中所遭遇之困難，本研究提出以工作為基礎之知識支援模式，預期達到有效知識彙集、遞送與分享之目的。

本研究首先提出系統化的工作相關知識評估機制，透過工作者間之協同合作以支援其資訊需求，並整合工作相關知識評估機制於知識支援系統中，以協助組織人員透過工作特徵檔擷取工作所需的知識。該工作相關評估機制，分析工作與知識物件之相關性並建置工作特徵檔(task profile)，以協助組織人員透過工作特徵檔擷取工作所需的知識物件，預期協助知識工作者從大量知識物件中有效獲取工作相關知識，克服工作執行中所遭遇之困難。在此基礎之上，我們更藉由知識工作者資訊回饋過程，提出修正工作特徵檔之方法外，依該工作特徵檔，提出工作社群網路分析與建構方法，並探討與評估知識工作者之間互動

所構成的社群網路如何促成知識遞送與分享。研究內容主要包括：(1)提出適性化的工作特徵模式，藉由工作相關回饋機制修正工作特徵檔，以描述知識工作者之動態性工作資訊需求；(2)提出工作同好群組分析法，依據工作者特徵檔分析知識工作者資訊需求之相似性，並建立工作社群網路。在我們後續的研究中，發現工作者對於知識密集性工作之資訊需求是動態的，會隨著時間與環境而演化改變。因此，有效之知識支援需提供適性化機制以依據工作者之動態需求提供相關知識；此外，工作之執行，常需逐步執行階段性任務以完成工作，而不同階段有不同之工作資訊需求。根據組織工作特性而由系統主動提供工作相關知識的相關研究並未考慮工作之階段性；因此，本研究進一步改良先前知識支援模式，提出工作階段性為基礎之工作相關知識支援模式與系統架構。研究內容主要包括：(1) 根據知識工作者不同時間點的工作特徵檔，運用相關係數分析法，偵測工作者目前之工作階段；(2)以組織之工作主題分類架構為基礎，分析知識工作者於工作執行中之主題變換情形，以判別知識工作者現階段資訊需求主題；(3)該模式依據作者之工作階段與需求主題之變換，進而調整其資訊需求特徵檔，提供符合工作階段性之相關知識。

本文並依所設計之知識支援模式而設計實驗，以驗證方法於提供知識支援之有效性。此外，並以物件導向方式實作以工作為基礎之知識支援系統，建構協同合作之工作環境，以提供有效的工作相關知識遞送與分享。該系統落實在一研究單位，藉由使用者滿意度回饋以評估系統之有效性。研究結果顯示該知識支援模式與系統能有效達成知識遞送並促進組織成員之知識分享。

關鍵字：知識管理系統、工作相關知識、編撰知識、工作相關評估、適性化工作特徵檔、知識遞送、知識分享、工作階段、知識支援平台

誌 謝

論文即將付梓，揮別博士生涯之際，心中有所感動、不捨與責任。當初所持有的理想，雖有未盡之憾，但學生仍會持有夢想，堅定樂觀向前邁進。

博士班期間，感謝指導教授劉敦仁老師，對怡瑾論文悉心的指導，並提醒我適時的沉澱，培養我獨立研究的態度。感謝口試委員羅濟群老師、楊千老師、陳彥良老師以及魏志平老師與指導教授，於口試期間對論文的建議與指正，以及對怡瑾生涯發展的關懷，謝謝老師們。此外，感謝亦師亦友的蕭敏雄老師在語文對學生的啟迪、感謝曾國雄老師不吝惜的指導與慈愛的笑容、洪永城老師、吳誠文老師對我的鼓勵與關懷。

每個擦身而過的緣分或深或淺，都給我無比的力量，怡瑾銘記在心。特別感謝博士班期間，嘉源與 *K*-support 的朋友們昆學、北晨、韋孝，在研究對我的勉勵與協助，並分擔我許多實驗室事務，謝謝你們的善解，願我們都能以開懷的氣度面對一切可能。感謝籌尹學姊、民新與聰洲學長、孟蓉、秋婷、文彥、政龍、之怡、春鋒、栩嘉、皇志…等可敬可愛的朋友們，大方的分享生活與研究的點點滴滴；多才多藝的博士班同學們昶瑞…等；教育學程的朋友心玫、金鳳…等；曾門的朋友宜中學長…等；以及交大的朋友們，謝謝你們貼心的關懷，幫助我渡過生活與研究的瓶頸。

對於陪我走過十年寒暑的男友俊佑，內心有無以言喻的感動，喜歡與你一起發現生活的無限可能與享受生命的喜悅!!最後，謹將此論文獻給我摯愛的外婆與家人，感謝您們賜與的福份，我會努力當一個可愛而堅強的孩子。

吳怡瑾 (nancy)

Phinally **D**one !!　于交大　2005/1/12

Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Research background and motivation

Deploying knowledge management systems (KMS) is an important strategy for enterprises to effectively managing business knowledge and gaining competitive advantage. The operations and management activities of enterprises are mainly based on tasks, in which organizational workers perform various tasks to achieve business goals [1][22][24][26]. Moreover, organizations try to maximize the use of knowledge assets to increase an organization's profitability and productivity with the support of contemporary knowledge management tools. KMS employs Information Technologies (IT), such as document management and workflow management to facilitate the access, reuse and sharing of knowledge assets within and across organizations [17][39]. That is, the critical role of Information Technologies (ITs) is to assist knowledge workers to reuse valuable knowledge assets to carry out business tasks successfully [6][17][46].

Generally, ITs focus on explicit and tacit dimensions in knowledge management activities [28][39]. The former, explicit knowledge management, is achieved by a codified approach. Intellectual content codified into explicit form can facilitate knowledge retrieval and reuse [89]. Knowledge repository, knowledge-based systems, and knowledge maps are the supports for knowledge storage, organization and dissemination. And a repository of structured and explicit knowledge, especially in document form, is a widely adopted codification-based strategy for managing knowledge in KMSs [17][81][89]. The latter, tacit knowledge management, puts emphasis on dialoging via social networks to facilitate knowledge sharing. Knowledge expert directories, yellow pages, communities of practices and talk rooms, support interpersonal communication for knowledge sharing [3][41]. Notably, empirical findings indicate that codifying intellectual content into a knowledge repository makes workers highly exploit existing organizational resources [29][49]. Accordingly, knowledge (information) retrieval is considered a core component to retrieve codified knowledge in KMS. An effective knowledge retrieval function can mitigate the difficulty of accessing knowledge items from a knowledge repository and support the operation of knowledge-intensive work in business environments [24][27].

In task-based business environments, an important issue of deploying KMS is providing task-relevant information (codified knowledge) to fulfill the information needs of knowledge workers during task execution. That is, effective knowledge management relies on understanding workers' information needs on tasks, for brevity, task-needs. Recently, the information retrieval (IR) technique coupled with workflow management systems (WfMS) was employed to support proactive delivery of task-specific knowledge according to the context of tasks within a process [1][2][23][24]. The KnowMore system maintains task specifications (profiles) to specify the process-context of tasks and associated knowledge items [1][2]. The Kabiria system supports knowledge-based document retrieval in office environments, allowing users to conduct document retrieval according to the operational context of task-associated procedures [15]. Context-aware delivery of task-specific knowledge thus can be facilitated based on the task specifications and current execution context of the process. Furthermore, a process meta-model specifying the knowledge-in-context is integrated with workflow systems to capture and retrieve knowledge within a process context [44]. Although providing an appropriate view for designing task-based knowledge support, the above works focus on specifying the process-context of tasks to support context-aware or process-aware knowledge retrieval, rather than on a systematic approach to construct task profiles. Moreover, the adaptation of profiles to track workers' dynamic information needs is not addressed.

For complex and knowledge-intensive tasks, the collaboration among knowledge workers may arise around common goals, problems and interests. Accordingly, contemporary KMSs rely on an effective approach to construct a community of practice to promote knowledge sharing. A community of practice consists of people who share common needs of information; hence, a community of practice is an effective approach to promote knowledge creation, transfer and sharing within or across organizations [3][13][18][41]. The Milk system supports informal communication and knowledge sharing for knowledge workers performing tasks in different work practices [3]. OntoShare, an ontology-based KMS, models the interests of users and provides automatic knowledge sharing in communities of practice with the aid of profiles [18]. Although user profiles had been employed to stimulate knowledge disseminations in communities of practice, they did not consider the identification of peer-groups with similar task-needs to form communities in the task-based business environment.

Furthermore, for knowledge-intensive tasks, such as research projects in academic institutions, and product development in R&D departments, it is more difficult to supply task-relevant knowledge during the progress of task execution. That is, works' information needs on task, for brevity, task-needs, generally change during the long run of task performance. Thus, the issues of identifying and tracking workers' current task-stages and task-needs topics, and adjusting their profiles during task performance deserve further exploration. To provide a more effective long-term knowledge support, we propose a task-stage knowledge support model that incorporates Information Filtering model with the identification of worker's task-stage and task-needs topics.

1.2 Research objectives and tasks

This dissertation mainly investigates the issues related to delivering and sharing codified knowledge from the perspective of business task. Major research objectives are listed below.

(1) Proactively delivering task-relevant knowledge to workers engaged in knowledge-intensive tasks.

- A task-relevance assessment approach is proposed to identify workers' information needs on task.
- A task-based knowledge support model is proposed to track and model workers' dynamic information needs on task. The proposed model also promotes knowledge sharing among knowledge workers.

(2) Enhancing task-based knowledge support model to provide effective knowledge support at different task-stages

- Developing a task-stage knowledge support model to provide task-relevant knowledge according to workers' dynamic task-needs at different task stages.
- Also, employing user modeling technique to identify worker's task-stage and task-needs topics of stages.

(3) Deploying a task-based *K*-Support portal to acquire, organize, and disseminate the organization's knowledge resources from the aspect of task.

- Providing a collaborative task-based workplace to facilitate knowledge retrieval and sharing among peer-groups.
- Delivering and sharing task-relevant knowledge to fulfill the workers' task-needs at various task-stages.

1.3 Contributions

The contribution of this dissertation is to achieve knowledge reuse and support from the perspective of knowledge-intensive task. That is, extracting, organizing, and disseminating relevant knowledge (codified knowledge) to fulfill the information needs of knowledge workers during task execution.

This work first proposes a novel *task-relevance assessment approach* to identify the knowledge worker's information needs on tasks. Rather than specifying task characteristics directly by knowledge workers, a systematic approach is desirable to create task profiles by analyzing retrieved documents and assessing the relevance among tasks. Note that historical task-related information items preserved in the knowledge repository, such as task descriptions and codified knowledge, are valuable knowledge assets to support task profile construction. The proposed approach generates task profiles by the collaboration of knowledge workers to analyze the relevance of tasks and codified knowledge. Task-based knowledge support is facilitated through providing knowledge workers relevant knowledge based on task profiles. Although this work does not consider the process-aspect and context awareness, as discussed in previously pilot studies [1][2][24][44], this approach can alleviate the problem of accessing needed knowledge items from vast amounts of codified knowledge.

Furthermore, methods of the adaptation of profiles to track workers' dynamic information needs are proposed in this work. The worker's dynamic task-needs can be analyzed based on the changes of workers' profiles during task performance. An *adaptive task-based profiling approach* is proposed to tackle worker's dynamic information needs on tasks. A task profile describes the key features of a task and is the kernel for discovering and disseminating task-relevant information to knowledge workers. This approach models the worker's task-needs based on feedback analysis, i.e. explicit or implicit feedback on knowledge items. In addition, this work not only considers the profiles of feedback items but also considers the profiles of relevant topics in the domain ontology. Note that we refer the domain ontology as the taxonomy of topics in our task-based problem domain. Different from traditional information filtering techniques with user profile, which only considered the profile of feedback items, the profile adaptation approach considers both the profiles of related tasks and the profiles of relevant codified knowledge to adjust the task profile.

For promoting knowledge sharing among workers, a *task peer-group analytical method* is proposed to identify task-based peer-groups according to workers' profiles, namely, task interests. The main characteristic of this method is that a fuzzy inference procedure is employed to infer the implicit and transitive relationships of knowledge workers based on task-needs. The proposed method can infer the implicit relationship among workers; even they did not provide feedback on the same knowledge items. With the aid of task-based profiles and peer-groups, the proposed $\mathcal{K}$-support portal can provide task-relevant knowledge and promote knowledge sharing among task-based peer-groups.

Moreover, according to our empirical investigation, knowledge workers engaged in knowledge intensive tasks (e.g., research projects in academic organizations, project management in firms, etc.) have different information needs during the long-term task performance. The Vakkari studies (2000, 2003), which focus on a user's information seeking activities during task performance (e.g., writing a proposal, completing a project, etc.), show that information needs vary according to different task stages. Therefore, we propose a knowledge support model based on task-stage to proactively deliver task-relevant knowledge. A *correlation analysis method* is proposed to identify a worker's task-stage (e.g., pre-focus, focus formulation, and post-focus task stages), and an *ontology-based topic discovery method* is proposed to determine a worker's task-needs topics of each stage. Consequently, the model can also be tailored to support long-term task performance.

Finally, we develop a collaborative task-based $\mathcal{K}$-support portal to facilitate knowledge reuse and to further promote knowledge sharing among peer-groups. The view of designing task-based knowledge support is the studies of context-aware or process-aware knowledge retrieval and knowledge delivery with the aid of user modeling. Details will be given in Section 3. Meanwhile, several experiments have been conducted to evaluate the effectiveness of the proposed knowledge support model based on task or task-stage in terms of precision and recall. The empirical system evaluation is also conducted to examine the effectiveness of the proposed system in terms of novelty and quality metrics.

1.4 Content organization

Fig. 1 illustrates the whole view of this work and the remainder of this work is organized as follows. The literature review is given in Chapter 2. Chapter 3 addresses the rationale to design task-based knowledge support system and presents the framework of the proposed system. Note that the tasks and functions of each module given in Fig. 1 are described in this section. Chapter 4 introduces the process of building the task-oriented repository, as depicted in the block one (B1) of Fig.1. The repository is designed for organizing and managing task-relevant information. In addition, a task domain ontology is structured to organize and classify knowledge items based on tasks.

The $\mathcal{K}$-support model and methods to provide task-based knowledge support with the aid of profiling technique are given in Chapter 5, 6, and 7. Note that the associated experiments to evaluate the effectiveness of the proposed methods are also carried out. Chapter 5 presents the proposed *task-relevance assessment approach* to identify the worker's information needs on tasks. The task-relevance assessment approach is designed to analyze the relevance of tasks and codified knowledge in the repository. Furthermore, a task profile is generated to support the proactive delivery of task-relevant knowledge. The assessment procedure is also given in the block two (B2) of Fig. 1. The lines with the numbers denote the assessment procedure. Next, Chapter 6 describes the proposed methods to disseminate and share task-relevant knowledge based on the generated profiles. The block three (B3) of Fig. 1 illustrates the main executed engines of Chapter 6 & 7. The *user behavior tracker* is an on-line module to capture workers' dynamic behaviors, including access behaviors on the task-based domain ontology and documents. The *task profile handler* uses *task-based profiling approach* to adjust workers' task profile to reflect workers' current task-needs (information needs on the target task). The *peer-group analyzer* employs *peer-group analytical method* for identifying task-based peer-groups with similar task needs based on task profiles. Details will be addressed in Chapter 6. Chapter 7 extends the task-based knowledge support model to provide effective knowledge support at different task-stages The t*ask-stage identifier* and *task-needs analyzer* are within the block three (B3), which are responsible for tracking the evolution of a worker's task-needs. Methods to identify worker's task-stage and task-need topics of stages are presented in this chapter.

Finally, the proposed $\mathcal{K}$-Support portal with associated system evaluation is presented in Chapter 8. Conclusions and future works are discussed in Chapter 9.

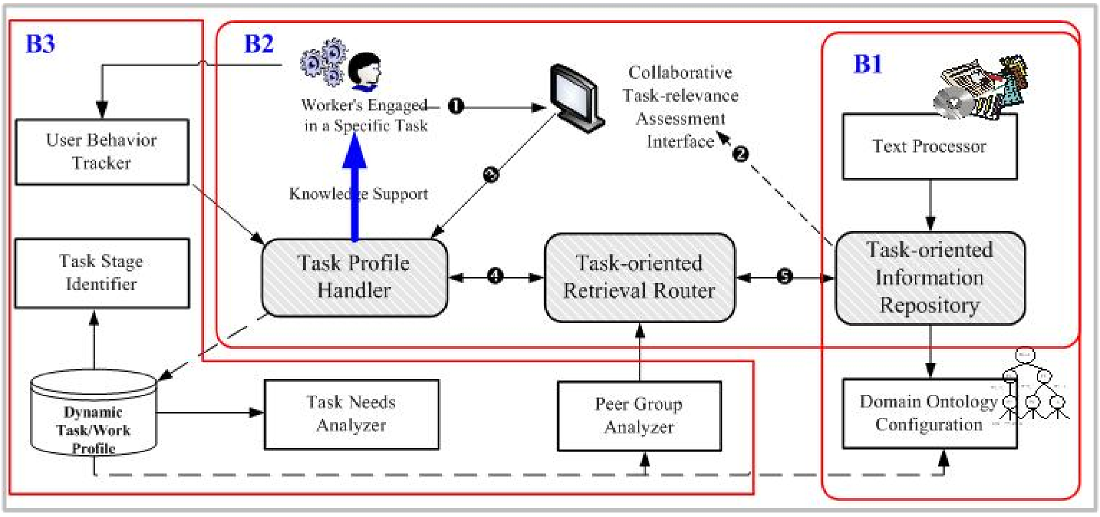


Fig. 1. Task-based knowledge support

Chapter 2 Related Work

2.1 Knowledge management in task-based working environment

2.1.1 Knowledge management systems and information technology

Knowledge Management (KM) is a cycle, sometimes repeated process, which generally includes creation, management and sharing activities. [17][26][28][55][82]. Organizations deploy Knowledge Management Systems (KMS) to maximize the effectiveness of knowledge assets in increasing organizational profitability and productivity [30][55]. Contemporary KMS employs Information Technologies (IT), such as document management and workflow management to facilitate the access, reuse and sharing of knowledge assets within and across organizations [17][39].

Generally, information technologies (ITs) mainly focus on two dimensions, explicit and tacit dimensions, to support knowledge management activities [11][29][39]. The former is achieved by codified approach. Intellectual content codified into explicit form can facilitate knowledge retrieval and reuse [12][89]. Knowledge repository, knowledge-based system, knowledge maps are the like to support knowledge storage, organization and dissemination [29][39][89]. The latter put emphasize on dialoging via social networks to facilitate knowledge sharing. Knowledge expert directories, yellow pages, communities of practices and talk rooms are the like to support interpersonal communication to rapid knowledge sharing [3][13][39][41]. Several researches classified the knowledge management practices based on the two dimensions.

According to Gray (2001a) empirical finding that the knowledge codified into knowledge repository make knowledge workers highly exposit existing resources within organization, whereas community of practices that provide informal personal communication can moderate to explore new possibility. Kankanhalli et al. (2003) pointed out those product-based firms in a high-volatility context are rely both codification and sharing approaches. Xerox, Microsoft, Hewlett-Packard are the examples. In summary, the critical role of ITs are to assist knowledge workers in fully and economic reusing valuable knowledge assets by decreasing the level of skills required in accomplishing the task successfully [28][39][49]. In addition, KMS with the aid of IT can assists workers in fully and economic reusing valuable knowledge assets to accomplish the objective of task successfully.

2.1.2 Task-based knowledge retrieval

The repository of structured, explicit knowledge, especially document form, is a codified strategy to manage knowledge [17][29]. However, with the growing amount of information in organizational memories, KMSs face the challenge to help users find pertinent and needed information. The information can be delivered in a specific context of business environments. The information retrieval (IR) technique coupled with workflow management systems (WfMS) was employed to support proactively delivery of task-specific knowledge according to the context of tasks within a process [1][24]. Furthermore, a process meta-model specifying the knowledge-in-context is integrated with workflow systems to capture and retrieve knowledge within a process context [44]. Despite the subtle difference among these works, they provide an appropriate view to achieve knowledge support based on tasks. Moreover, knowledge retrieval is also considered a core component in task-based business environment to access knowledge items in knowledge repository [24][27].

Herein, we categorized the task-based knowledge management work from two perspectives: one is knowledge delivery with the aid of user modeling and the other is context-based proactively delivery knowledge. The perspective is departure from the points of process complexity and knowledge intensiveness [21]. Based on the above points, for classes of business process are derived which are low (or high) business process and weak (or strong) knowledge intensity. In the following, the related works of task-based knowledge management will be given according to the classifying of business process.

Task-based knowledge delivery with the aid of user modeling: This kind of knowledge management framework put emphasizes on codified (e.g., documents) knowledge retrieval and delivery in supporting workers' day-to-day tasks operation. Translating users' information needs into compromised queries is not an easy work [75]. Most systems rely on Information Retrieval (IR) techniques to access organizational codified knowledge. The technique of Information Filtering (IF) with a profiling approach to model users' information needs is an effective approach to proactive delivering relevant information to users. The technique has been widely used in the areas of Information Retrieval and Recommender Systems [31][52][58]. The profiling approach has also been addressed by some KMSs to enhance knowledge retrieval and further promote knowledge sharing among project-based or interesting groups [1][2][3][18]. Accordingly, the techniques of information filtering with intelligent agent-based architecture are commonly adopted in this type of framework to streamline the knowledge delivery from internal or external knowledge repositories [73][88]. Notably, a promising user modeling method, in which the system delivers the relevant information to the user profile is demanded in this type of knowledge support [7][71]. The idea of cooperative agent architecture has been proposed to achieve task-based Information filtering within work process [19]. Three types of cooperating agents: process agents, document warehouse agents and retrieval agents are designed for evaluating if the retrieved documents are relevant to the workers' tasks at hand. Furthermore, a *CodeBroker* system is proposed for supporting software developer to reuse the organizational program components repository properly [88]. Similarly, the information filtering with user modeling and agent-based techniques are applied in the system for making delivered information relevant to the task-at-hand and personalized to the worker's information needs. The task-based knowledge delivery with the aid of user modeling is quite suit applied in knowledge intensive task due to it has capability to model worker's task needs and individual needs based on user modeling technique. The chief defect of this framework is that it generally cannot proper incorporate the contextual information of business task into the user profile.

Context-based proactively knowledge delivery: The information can be delivered in a specific context of business environments. To this end, KMSs increasingly emphasize the organization of all the possible task-specific knowledge by supporting context-aware knowledge access and retrieval [1][5][44]. The Kabiria system supports knowledge-based document retrieval in office environments by allowing users to conduct document retrieval according to the operational context of task-associated procedures [15]. Furthermore, a process meta-model specifying the knowledge-in-context is integrated with workflow systems to capture and retrieve knowledge within a process context [44]. That is, context becomes an impartment component that can be utilized for improving the understanding of relevant knowledge of business task within the KMS. Recently, the knowledge context model is even proposed to support the collaborative work of virtual teams by utilizing the contextual information [4]. Furthermore, acquiring and disseminating role-relevant process views was considered in workflow environments [72]. Alvarado et al. (2004) also proposed acquiring and organizing corporate memory from the perspective of role/job position, in which an Organizational Memory is modeled by adopting UML/XML to specify the ontologies for organization positions, tasks, and application domains. The context-based knowledge delivery model is quite suit applied in knowledge intensive and non- routine task due to it has knowledge context model to capture or utilize the business process context for supporting task execution. Furthermore, it can even support the operation of business process with high process complexity. However, the kind of knowledge support model still lacks in learning capability to support real time context sensitive knowledge delivery till know. That is, besides understanding the work context of the given task, the model also needs to learn and response the worker's task-needs in the real time.

2.1.3 Knowledge sharing in community of practices

For complex and knowledge-intensive tasks, the collaboration among knowledge workers may arise around common goals, problems and interests. Domain experts or experienced workers who hold valuable tacit knowledge play important roles in assisting knowledge workers to accomplish business tasks [51]. The ultimate goal of KM is to enable innovative activities by promoting collaboration or communication among knowledge workers in organizations [26][84]. Collaboration may take place in a formal group such as a business project or in an informal group such as a community of practice. A community of practice consists of people who share common needs of information; hence, a community of practice is an effective approach to promote knowledge creation, transfer and sharing within or across organizations [3][13][18][41]. Although user profiles had been employed to stimulate knowledge disseminations in communities of practice, they did not consider the identification of peer-groups with similar task-needs to form communities in task-based business environments.

2.2 Text mining technique for codified knowledge management

2.2.1 Information retrieval in vector space model

The key contents of a codified knowledge item (document) can be 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 [67]. The *term transformation* steps, including case folding, stemming, and stop word removing, are conducted during text pre-processing [7][60][65][83] The *term weighting* then is employed to extract the most discriminating terms [67]. Let d be a codified knowledge item (document), and let $\vec{d} = <w(k_1, d), w(k_2, d), \ldots, w(k_n, d)>$ be the feature vector of d where $w(k_i, d)$ is the weight of a term k_i that occurs in d. Notably, the weight of a term represents its degree of importance to represent the document (codified knowledge). The well-known *tf-idf* approach is often used for *term (keyword) weighting*. The approach assumes that terms with higher occurrence frequency in a document and occurring in fewer other documents are better discriminators to represent the document. Let the term frequency $tf(k_i, d)$ be the occurrence frequency of term k_i in d, and let the document frequency $df(k_i)$ represent the number of documents that contain term k_i. The importance of term k_i to a document d is proportional to the term frequency and inversely proportional to the document frequency, which is expressed as Eq. 2.1.

$$w(k_i, d) = \frac{1}{\sqrt{\sum_i \left(tf(k_i, d) \times \log(N/df(k_i))\right)^2}} tf(k_i, d) \times \log \frac{N}{df(k_i)} \qquad \textbf{(2.1)}$$

where N is the total the number of documents. Notably, the denominator in the right side of Eq. 1 is a normalization factor to normalize the weight of term.

Similarity measure: The cosine formula is a widely used similarity measure to assess the degree of similarity between two items x and y by computing the cosine of the angle between their corresponding feature vectors $\vec{x}$ and $\vec{y}$, which is given by Eq. 2.2.

$$sim(x, y) = cosine(\vec{x}, \vec{y}) = \frac{\vec{x} \bullet \vec{y}}{|\vec{x}||\vec{y}|} \qquad \textbf{(2.2)}$$

The degree of similarity is higher if the cosine similarity is close to 1.0.

Each document or query can be represented as a document or query feature vector in a vector space model. Let $\vec{d}_j$ represent a document vector of a document d_j and let $\vec{q}$ be a query vector of a query q. The similarity between a document d_j and a query q, $sim(d_j, q)$, can be calculated as the cosine of the angle between the two vectors $\vec{d}_j$ and $\vec{q}$, namely $cosine(\vec{d}_j, \vec{q})$.

2.2.2 Relevance feedback techniques

Relevance feedback effectively improves search effectiveness through query reformulation. Various studies have demonstrated that relevance feedback applied in the vector model is an effective technique for information retrieval [63][68]. Eq. 2.3 and 2.4 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 $\vec{q}$ [7].

$$\text{Standard\_Rocchio: } \vec{q}_m = \alpha\vec{q} + \beta\frac{1}{|D_r|}\sum_{\forall d_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_n|}\sum_{\forall d_j \in D_n}\vec{d}_j \tag{2.3}$$

$$\text{Ide\_Dec\_Hi: } \vec{q}_m = \alpha\vec{q} + \beta\sum_{\forall d_j \in D_r}\vec{d}_j - \gamma \max_{irrelevant}(\vec{d}_j) \tag{2.4}$$

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 $\max_{irrelevant}$ returns the most irrelevant document. The two methods produce similar results [7]. Most studies suggest that the information of relevant documents is more important than that of irrelevant documents [32][68].

2.3 User modeling for information filtering

2.3.1 Researches of information filtering to support information needs

Information retrieval and information filtering technologies applied in document management systems are generally the first pace of knowledge management initiatives, since textual data such as articles, reports, manual, know-how documents and so on are treated as valuable and explicit knowledge within organizations [55]. Information retrieval and information filtering are considered as the core techniques to achieve knowledge retrieval. In addition, information retrieval provides not only text processing technique, but also document classification technology to help organizations collect and process documents to achieve the goal of knowledge reuse [70]. With the aid of information filtering, it not only reduces the problem of information overloading but also provides relevant and needed information to users to accomplish their tasks.

Information filtering (IF) systems are commonly personalized to support long-term information needs of a particular user or a group of users with long-term information needs [53][54][80]. IF systems are similar to conventional information retrieval (IR) systems. The IR system mainly focuses on facilitating user's short-term information needs, e.g. generally expressed information needs in a single search session. However, the IF system relies on the support of the kernel technology of IR, but it puts emphasis on methods to maintain and learn user profiles to support long-term information services [7][9][80].

IF stresses on maintaining a promising user profile, in which the system delivers the relevant information to the user profile [7][71]. Various methods for learning user interests or preferences from text documents or Web pages have been proposed [7][8][10][52][53][54][58]. The well-known methods in Information Retrieval or Information Theory are modified and then employed to model user's dynamically changed interests, for example, Rocchio algorithm, information gain theory, Bayesian classifier. Notably, all these learning algorithms require relevance feedback collection process, either explicit feedback (where system collects user linguistic ratings) or implicit feedback (where system monitors user access behavior). The IF system learns the users' current task-needs from the feedback on the supported information, and updates the model for future information filtering. Such kind of learing method can maintain the user profiles once the system received the feedback; therefore, the learning method is regarded as the incremental learning technique.

The IF technique is realized in many real-world applications, for example: e-mail-filtering systems [53], personalized online newspaper [10], adaptive Web page recommendation service [8], and on-line academic research paper recommendation [52]. Accordingly, IF technology is acknowledged to be an effective way to reduce the information overload and provide personalized information [30][35]. Although IF systems provide proper profiling method to learn user's dynamic needs/interests; however, most of existing systems do not consider integrating the user's information needs with the progresses of task performance. A promising profile modeling approach considering the characteristics of task stage and user's current information needs is more demanded in task-based business environments.

2.3.2 User modeling technique to support knowledge-intensive tasks

The characteristic of knowledge retrieval activity in working environment is that the worker's information needs is associated with the executing task at hand. Meanwhile, a knowledge-intensive task consists of levels of progressively smaller subtasks to achieve the main task goal. That is, when the worker confronts with the task, there is a gap between the worker's knowledge about the task and the perceived requirements of tasks. The gap is the information need and results in information seeking activities [14]. Generally, a worker uses documents to understand a task, solve the encountered problem, or result in another search behavior for finding a solution. Accordingly, several empirical studies focus on how documents are selected and used by workers during task performance. A well-known longitude project has been conducted to investigate a cognitive model of document use during a research project [78][79]. The study models document use as a decision-making process where decisions may occur at three points or stages during a research project, which are selecting, reading, and citing.

Several empirical studies concentrated on discovering and analyzing the growth in students' or scholars' understanding of their own assigned tasks during conducting an actual research project [43][50][76][77][78][79]. The Kuhlthau's study (1993) [43]is to observe people involved in information seeking over a period of time. Six stages were identified in his empirical study from the students' description of their experience; these stages match the phases in the process of construction. The Vakkari (2000) studies concentrated on the user's information seeking activities during the progress of task performance (e.g. writing a proposal, completing a project and the like). The Vakkari study is based on the Kuhlthau's model to connect the research of information seeking activities to the pre-focus, focus-forming and post-focus stages of the process [76][77]. The empirical studies reveal that users' information needs will vary at different task stage. For example, the types of information needs may vary from general information to specific information, and the choice of search terms is varied from broader terms to related terms. That is, a worker's information needs and information-seeking processes depend on worker's progresses of task performance, or task stages, specifically.

The characteristic of knowledge retrieval activity in working environment is that the worker's information needs are associated with the executing task at hand. Meanwhile, a knowledge-intensive task consists of levels of progressively smaller subtasks to achieve the main task goal. Therefore, the concept of task stage in information seeking studied can support this work for providing task-relevant knowledge more precisely. And a promise knowledge support model to reflect workers' current task-needs and task-stage is a critical issue deserved exploration.

Chapter 3 Task-based Knowledge Support

3.1 Rationale to design task-based knowledge support

The proposed work focuses on providing knowledge support for knowledge-intensive tasks within organizations. Examples of knowledge-intensive tasks include thesis works and research projects in academic organizations, project management in firms, research work and product development in R&D departments, and the like. In such task-based environments, reusing knowledge assets extracted from historical task executions is the key to providing effective knowledge support for conducting tasks.

Historical codified knowledge, i.e. experiences and know-how extracted from previous task executions, provides valuable knowledge for conducting tasks. For example, effective project management can benefit from KMS by referring similar projects to acquire best practice, lessons learned, working experiences, or knowledge resources. Research task innovation is generally based on previous research achievements. A knowledge repository that preserves the experience and knowledge of previous work (research task) is important to provide effective knowledge support for research tasks. However, with the increasing amount of information in the organizational memory (OM), contemporary KMS faces challenge to assist organizations acquire, organize and manage knowledge. Thus, delivering relevant historical codified knowledge to workers for accomplishing tasks at hand is also a challenging work deserves exploration. This work sought to tackle the challenges from the perspective of business task.

3.1.1 Task-based organizational environment

"Mary is a new worker of an industry analyzer in a project management institution. She is assigned to a survey task, "the opportunities of sensor network in healthcare", and need to write a proposal. Since Mary is a novice of sensor network, she faced the problem to understand the assigned task. She wants to find task-related expert or colleague to solve the encountered problem or guide him to the right direction while understanding the perceived task. Unfortunately, workers who have relevant knowledge are busy for the business projects. Hence, Mary comes up with the idea to find the possible solutions from the document management system or information

repository in the organization. However, tremendous amount of data frustrated Mary. That is, it is hard for Mary to have a clear view of information structure or taxonomy of the document management system or information repository."

The situation generally happens in the organization, especially in IT or MIS department of industry, or the industry analyst in project management institution. When a worker in an organization has information needs of the executing task, he/she might need the knowledge support to accomplish the task. Naturally, the worker may seek someone who has met this problem or has done similar experiences before. Otherwise, the worker may also try to find the relevant codified knowledge from the organizational repository. Thus, if knowledge resources in an organization are acquired, organized via the view of business tasks, workers could get more effective knowledge support.

3.2 Framework of task-based knowledge support

Figure 2 illustrates the system framework of the proposed task-based knowledge support based on profiles to facilitate task-based knowledge delivery and sharing. Participants include knowledge workers engaged in specific tasks and domain experts in specific subjects. The system comprises four main modules, namely *task-oriented information repository*, *task profile handler*, *task-needs evolution*, and *task-oriented information service router*.

Task-oriented information repository. The task-oriented information repository is designed for organizing and managing task relevant information. Building a proper repository to acquire and disseminate knowledge items is a key strategy for managing knowledge in the contemporary KMS. Information items indexed by proper concepts and categories can provide knowledge workers with meaningful access to organize intellectual content. Task-oriented repositories are constructed with support from category schema to effectively utilize codified knowledge. Such a repository stores codified knowledge corresponding to task execution, and contains three main databases, including the *document-indexing database, task corpus, and task categorization database*. The *document-indexing database* stores task relevant documents indexed using the inverted file approach. Meanwhile, the *task corpus* stores the key profile of each existing task. An ***existing-task*** is a *historical task* accomplished within the organization. Task corpus is used to describe the key

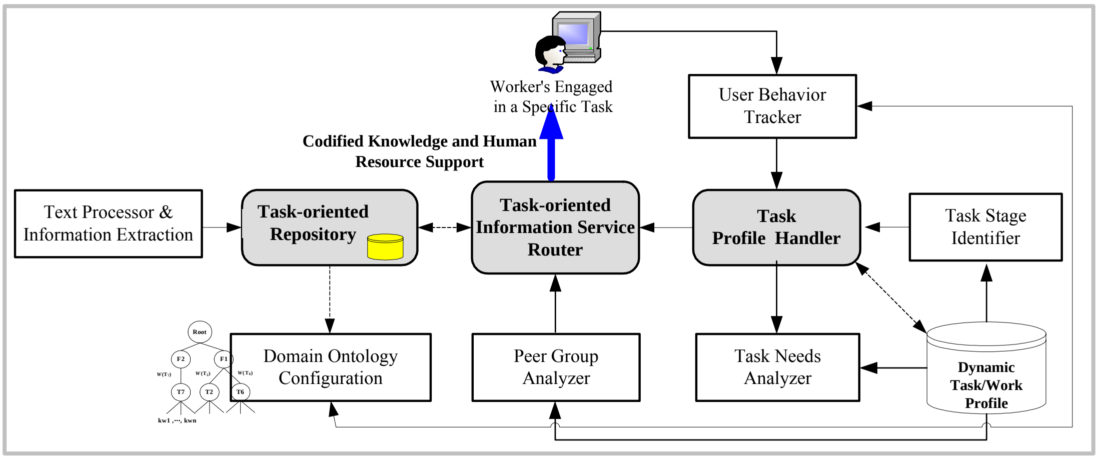


Fig. 2. Framework of task-based knowledge support

subjects of an existing task, and is expressed as a feature vector of weighted terms. Section 4.1 details the extraction of task corpus of an existing task, which is derived by extracting the weighted terms from textual documents generated and accessed by the task. Moreover, the ***task categorization database*** records the relationships of existing tasks and categories, namely, the relevance degrees of existing tasks to categories. The *task categorization database* is used to support the operation of identifying referring tasks based on their similarity to the executing task derived using the relevance degrees of tasks to the categories. Moreover, tasks with similar subjects are grouped into fields. The repository is the knowledge base for task-based knowledge support. Details are discussed in Chapter 4.

Task profile handler. The *task profile handler* provides mechanisms such as profile creation, adjustment, integration and profile adaptation to conduct profile management. Two kinds of profiles, feature-based task profile and topic-based task profile, are maintained to model workers' information needs on the target task at hand.

- **Feature-based task profile** describes the key features of a task and is the kernel for discovering and disseminating task-relevant information to knowledge workers.
- **Topic-based task profile** models a worker's information needs on the target task, and is represented as a set of relevant tasks or fields of the target task with associated relevance degrees.

Workers' information needs may change during the progress on performing the target task. The *user behavior tracker* is an on-line module to capture workers' dynamic behaviors, including access behaviors on the task-based domain ontology and relevance. The *profile handler* uses an adaptive task-based profiling approach to adjust workers' profiles. The *peer-group analyzer* employs a task-based peer-group analytical method to identify peer-groups with similar task needs (information needs on the target task) based on work profiles. Details will be addressed in Chapter 6.

Task-needs evolution. The t*ask-stage identifier* and *task-needs analyzer* are within this module, which are responsible for tracking the evolution of a worker's task-needs. The profiles are employed as indicators for *task-stage identifier* and *task-needs analyzer* to model the worker's task-needs of target task. Herein, worker's task-needs are modeled as the topics nodes in domain ontology (DO) at different abstraction level which are relevant to the on-going task. The DO is a multi-level structure and each node in the DO represents a research topic in our application domain, as given in the Figure 3 of Chapter 4. The *task-stage identifier* is responsible for analyzing and determining worker's task stage based on the changes of the task profile over time. The *task-needs analyzer* is responsible for tracking the worker's access behavior over a period of time. The access behavior is analyzed based on the DO to discover worker's task-needs on specific topics. Details are discussed in Chapter 7.

Task-oriented information service router. The router helps knowledge workers gather appropriate information from the task-oriented repository and task-based peer-groups. The router fetches task-relevant information according to the worker's task profile. Moreover, each worker has his/her own view of task-relevant information, namely, personalized ontology, which is derived from his/her work profile on the target task and is organized according to the domain ontology. Knowledge sharing from other peer-group members is derived by retrieving each peer-group member's personalized ontology. Details are addressed in Chapter 6.

Chapter 4 Task-Oriented Information Repository: Managing Codified Knowledge

4.1 Task-oriented information repository

To organize and manage task-relevant information, the repository is constructed with support from domain ontology (i.e., topic taxonomy) to effectively utilize codified knowledge. This session discusses the issue of managing codified knowledge with the support from category scheme.

Categories representing the main subjects of organizations are defined to organize tasks and codified knowledge. Task corpus (feature vector of weighted terms) describing the key subjects of existing task can be constructed by extracting the weighted terms from textual documents. The *task categorization database* records the relevance degrees between existing tasks and categories based on the result the proposed task categorization model. The *task categorization database* is used to support the operation of identifying referring tasks based on their similarity to the executing task derived using the relevance degrees of tasks to the categories. Identifying a small subset of existing tasks as referring tasks can help knowledge workers conduct further task-relevance assessment without reviewing all existing tasks. This chapter illustrates two essential phases in constructing a task-oriented information repository: extracting task corpus from textual data gathered during task execution and deriving the relevance degrees between existing tasks and categories.

4.1.1 Extracting task corpus

The task corpus of a task t_r is represented as a feature vector of weighted terms (keywords) derived by analyzing the set of documents generated and accessed by t_r. Each document d_j is pre-processed and represented as a feature vector $\vec{d}_j$. The centroid approach is employed to derive the feature vector of a task by averaging the feature vectors of documents generated and accessed by the task. Let D_{tr} denote the set of documents that are generated and accessed by task t_r. Furthermore, the task corpus (feature vector) of task t_r is defined as the **centroid** vector $\vec{t_r}$ which is the vector obtained by averaging the feature vectors of documents in D_{tr}. Eq. 4.1 defines the **centroid** vector $\vec{t_r}$. The weight of a term k_i in $\vec{t_r}$ is represented as $w(k_i, t_r)$.

$$\vec{t}_r = \frac{1}{|D_{t_r}|} \sum_{d_j \in D_{t_r}} \vec{d}_j \quad \textbf{(4.1)}$$

4.1.2 Task categorization model

Existing tasks are categorized based on fuzzy classification, and thus they may belong to more than one category. Fuzzy classification extends the traditional crisp classification notation to associate each object in every category with a membership function so that each object can belong to more than one category (Zadeh, 1965). The *task categorization database* records the relationships of existing tasks and categories, namely, the relevance degrees of each existing task to categories. The relevance degree between a task and a category indicates the strength that the task belongs to the category. The relevance degrees between categories and existing tasks are calculated based on the similarity measures between feature vectors of categories and existing tasks. The feature vector of a category is also expressed as a vector of weighted terms, which represents the main subjects of a category.

The categorization procedure includes the step of deriving the feature vectors of categories and the step of deriving the relevance degrees between existing tasks and categories.

Deriving the feature vector of each category: Experts predefined a set of categories to represent the main subjects within the organizational domain, such as "Text Mining", "Knowledge Management", etc. The seed-based approach is then applied to generate the feature vectors of categories. Experts select some existing tasks which represent a category. The selected tasks are called the seed tasks of the category. Once the seed tasks have been decided, a centroid vector can be derived from the corpora (feature vectors) of the seed tasks to describe the category. The centroid vector of each category is derived by averaging the feature vectors of corresponding seed tasks.

Let X denote a set of categories, $X=\{c_1, c_2, \cdots, c_m\}$, and let T_{cj} represent the set of seed tasks of category c_j. Let $\vec{c}_j^{\,c}$ be the **centroid** vector derived from the task corpora (feature vectors) of seed tasks of c_j. The centroid weight of term k_i in $\vec{c}_j^{\,c}$, $w(k_i, \vec{c}_j^{\,c})$ is derived as Eq. 4.2.

$$w(k_i, \vec{c}_j^{\,c}) = \frac{1}{\left|T_{C_j}\right|} \sum_{t_r \in T_{c_j}} w(k_i, t_r) \qquad \textbf{(4.2)}$$

The centroid vectors are used as the initial feature vectors of weighted terms to represent categories. The initial centroid weight of a term represents the degree of importance of the term in a category without considering its importance in other categories, namely its discriminating power to distinguish categories. The weight of a term is further adjusted by considering the discriminating power of the term. For example, a higher weight term denotes that it is a more representative and important concept of the category. However, some terms with a high weight in a category may also have high weights in other categories. Such terms may be common terms, even though they have high weights in categories, which are not discriminating enough to represent each category. To decrease the weight of this kind of common term, we use the probability distribution of terms across categories as a factor to discriminate the categories. Consequently, the weight of a term in a category is adjusted by multiplying the initial centroid weight of the term with the probability distribution of the term appearing in the category.

Let $\vec{c}_j$ be the feature vector of category c_j which denotes the key concepts of c_j, and let $w(k_i, c_j)$ be the weight of term k_i in category c_j. Then $w(k_i, c_j)$, the importance of term k_i in representing category c_j, is proportional to the centroid weight of term k_i and the probability distribution of term k_i appearing in category c_j, which is expressed as Eq. 4.3. Notably, $P(k_i, c_j)$ is the probability distribution of term k_i appearing in category c_j, which is computed according to the distribution of centroid weights of term k_i across categories.

$$w(k_i, c_j) = \frac{1}{\sqrt{\sum_i \left(w(k_i, \vec{c}_j^{\,c}) \times P(k_i, c_j)\right)^2}} w(k_i, \vec{c}_j^{\,c}) \times P(k_i, c_j)$$

$$where\ P(k_i, c_j) = w(k_i, \vec{c}_j^{\,c}) \Big/ \sum_{j=1}^{m} w(k_i, \vec{c}_j^{\,c}) \qquad \textbf{(4.3)}$$

where m is the number of categories. Notably, the denominator in the right side of Eq. 4.3 is a normalization factor to normalize the weight of term.

Deriving the relevance degrees of existing task to categories: Once the feature vector of weighted terms for each category has been extracted, we can derive the relationship (relevance degree) between categories and existing tasks based on the cosine measure. The membership grade (relevance degree) of task t_r to category c_j, $\mu_{cj}(t_r)$, can be calculated as the cosine of the angle between two vectors, $\vec{t}_r$ and $\vec{c}_j$, namely *cosine* ($\vec{t}_r$, $\vec{c}_j$). The relevance degree between a task and a category indicates the strength that the task belongs to the category. The relevance degrees of task t_r to the *m* categories can be modeled as a vector $\overrightarrow{t_r^C}$ characterized by the membership grades of t_r to the categories, as expressed in Eq. 4.4.

$$\overrightarrow{t_r^C} =< \mu_{c_1}(t_r), \mu_{c_2}(t_r), .., \mu_{c_m}(t_r) > \quad \textbf{(4.4)}$$

The task categorization database records the fuzzy classification result. Each task t_r is associated with its membership grades (relevance degrees) to categories. Notably, the *task categorization database* is used to support the operation of proposed two-phase task-assessment approach. The details are described in Section 5.1.

4.2 Domain ontology formalization

The domain ontology, a shared conceptualization of a specific domain, is often used to specify the working domain of an organization [57]. Organizing knowledge items into ontological structure based on the domain ontology is promising to support knowledge retrieval in business environments [25]. In this project, we refer the domain ontology to a classification structure of tasks stored in the knowledge repository. Specifically, the domain ontology (DO) is *a simple topic taxonomy that is structured into four levels, including categories, fields, tasks and knowledge items*, as shown in Figure 3.

Categories representing the main subjects of organizations are pre-defined to organize tasks and codified knowledge. Tasks with similar subjects are grouped into fields. This work labeled name of fields according to the schema of ACM Computing Classification Systems (1998). Notably, the relevance degrees to categories represent the subjects of a task, as addressed in Section 4.1. The similarity between tasks can thus be calculated based on their relevance degrees to categories. Based on the fuzzy relationship matrix ***R***, similar tasks are grouped together to form a field, as follows. A threshold value, *thres*θ, is defined to transform the fuzzy relation matrix ***R*** into a binary relation matrix ***B***. The threshold value is determined by the max-min operation, as shown in Eq. 4.5.

$$thres\theta = \max(\mu_{\min}(t_1), \mu_{\min}(t_2), \cdots, \mu_{\min}(t_k))$$
$$where \ \ \mu_{\min}(t_r) = \min(\mu_{C_1}(t_r), \mu_{C_2}(t_r), \cdots, \mu_{C_m}(t_r)) \quad \textbf{(4.5)}$$

According to Eq. 4.6., the fuzzy relation matrix ***R*** is transformed into a binary relation matrix ***B***.

$$\mu_{C_i}(t_r) = \begin{cases} 1 & \mu_{C_i}(t_r) > thres\theta \\ 0 & \mu_{C_i}(t_r) \le thres\theta \end{cases} \quad \textbf{(4.6)}$$

Tasks that have the same relationship with respect to each category in ***B***, are similar tasks to be grouped into a field labeled by a field name. The result generates a l-by-k field-to-task relation matrix $\boldsymbol{F} = [f_j(t_r)]$ such that $f_j(t_r)$ is one if task t_r is grouped into field f_j; and is zero otherwise; where l denotes the number of fields.

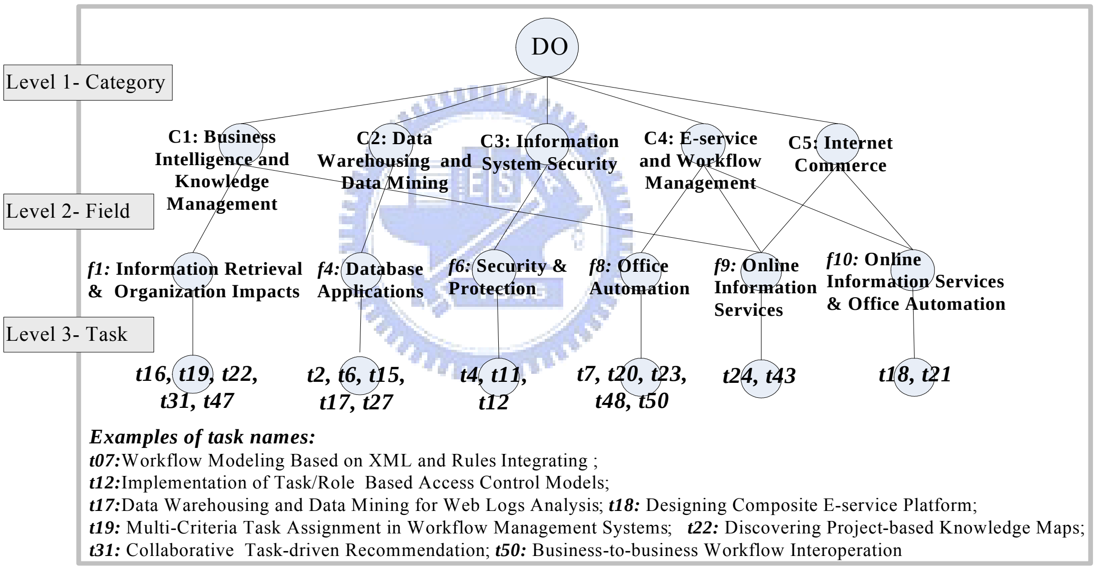


Fig. 3. Example of domain ontology

Chapter 5 Collaborative Task-Relevance Assessment

In knowledge-intensive task-based business environments, a pertinent issue in deploying KMS is providing task-relevant information (codified knowledge) to fulfill the information needs of knowledge workers during task execution. Accordingly, an effective knowledge retrieval function is more demanded to mitigate the difficulty of accessing knowledge items from a knowledge repository. Knowledge workers can access codified knowledge by submitting a query. However, workers may have difficulty in precisely expressing their information needs (as queries), particularly in conducting a new task. This work overcomes this problem by constructing task profiles to model workers' task-needs. Moreover, a collaborative relevance-assessment approach is employed to help novices with the aid of experts and collaborative workers for streamline the profile constructing process.

This chapter first introduces the concepts of collaborative relevance assessment and presents the assessment process for generating task profile. Next, method of conducting two-phase collaborative relevance-assessment for generating task profiles is given. A two-phase collaborative relevance assessment is used to discover a set of referring tasks that are relevant or irrelevant to the executing task. The proposed approach employs collaborative assessment with fuzzy linguistic evaluation to generate task profiles based on the task corpora of existing tasks. The task profile of an executing-task (or a new task) is initially derived via analyzing the task contents, or alternatively using the corresponding task corpus. Three experiments were performed to evaluate the assessment and retrieval effectiveness based on the proposed methods. Experiments using a real application domain were carried out for conducting research tasks in a research institute laboratory. Finally, the effectiveness of collaborative relevance assessment is discussed.

5.1 Preliminary concepts and term definition

5.1.1 Preliminary concepts

Primary concepts of collaborative task-relevance assessment are listed below.

- Existing tasks may have different degrees of relevance to the executing task. A fuzzy linguistic approach is used to evaluate task relevance and codified knowledge by using linguistic terms such as "*low*" or *"high"* to express the perception of "*Relevance*". Fuzzy linguistic approach is an approximate technique for modeling human thinking, and provides easier assess in evaluating qualitative problems [91]. Users express their evaluation in linguistic terms instead of numbers. The proposed task-relevance assessment provides a systematic and natural way to analyze the relevance of tasks and codified knowledge in the repository.

- Domain experts or experienced workers with valuable tacit knowledge play an important role in helping knowledge workers solve problems or make decisions [23][26][27]. For complex and knowledge-intensive tasks, collaboration among knowledge workers and experts is often necessary for more effective knowledge dissemination. The proposed mechanism employs a collaborative assessment approach to help knowledge workers, especially novices, evaluate the relevance of tasks and codified knowledge via collaboration of colleagues and domain experts. Accordingly, the proposed system can provide more effective knowledge support through knowledge sharing.

- Conducting relevance assessment on a large number of existing tasks may create a burden for workers and influence the assessment result. The proposed mechanism reduces the number of tasks to be assessed via discovering a set of referring tasks from existing tasks to assist workers in conducting task-relevance assessment. Identifying a small subset of existing tasks as referring tasks can help knowledge workers conduct further task-relevance assessment without reviewing all existing tasks. The referring tasks are selected based on their similarity to the executing task using the relevance degrees of tasks to the categories.

- The referring tasks form the basis to extract task-relevant knowledge for the executing task. Once the referring tasks are identified according to the assessment, a modified relevance feedback (RF) technique is used to derive the task profile of the executing task based on the relevance of referring tasks. Relevance feedback is a well-known technique in information retrieval for improving search effectiveness by automatic query reformulation [7][63][64][83]. Notably, a task profile specifies the key subject of an executing task, and can be used to retrieve relevant codified knowledge from the knowledge repository.

5.1.2 Term definition

This section lists the definitions of terms.

- **Task:** The task is the fundamental unit in business. This work broadly refers to a task as a unit of work in organizations, such as a project, piece of research work, or activity. A task denotes either an *existing-task* or an *executing-task*.
- **Existing-task:** An *existing-task* is a *historical task* accomplished within the organization.
- **Executing-task:** An *executing-task* is the *target task* that the knowledge worker currently conducts at hand.
- **Referring-task:** *Referring tasks* are existing tasks selected to assist workers in conducting task-relevance assessment. Identifying a small subset of existing tasks as referring tasks can help knowledge workers conduct further task-relevance assessment without reviewing all existing tasks. The referring tasks are selected based on their similarity to the executing task derived using the relevance degrees of tasks to the categories.
- **Task profile**: A *task profile* specifies key subjects of an executing task, and is constructed to model the information needs of knowledge workers during task execution.
- **Task corpus:** The *task corpus* is used to describe the key subjects of an existing task, and is expressed as a feature vector of weighted terms.
- **Task categorization database:** The *task categorization database* records the relationships of existing tasks and categories, namely, the relevance degrees of existing tasks to categories.
- **Term:** A *term* means a single word (e.g., “knowledge”, “management”) within the text of the document, and all words that appear in the set of document collection are dimensions to our term vectors.

- **Weighted terms:** Generally, the system added weight assignments to provide distinctions among the terms, e.g., (knowledge, 0.5), (management 0.8). The well-known *tf-idf* approach is generally used for *term (keyword) weighting.*
- **Term frequency:** *Term frequency* factor measures the frequency of occurrences of the terms in the document or query.
- **Fuzzy classification:** *Fuzzy classification* extends the traditional crisp classification notation to associate each object in every category with a membership function so that each object can belong to more than one category.
- **Fuzzy linguistic approach:** A technique for approximating human perception, and provides easier assess to qualitative problems. Users can express their evaluation by linguistic terms instead of numbers.
- **Fuzzy number:** A fuzzy number $\tilde{Z}$ is a fuzzy set defined on the real set $\mathbb{R}$. Fuzzy numbers can be used to represent linguistic terms, such as each linguistic term of the "Relevance" variable in this work.

5.2 Process of task-relevance assessment

Knowledge workers can access codified knowledge by submitting a query. However, such workers may have difficulty in precisely expressing their information needs (as queries), particularly in conducting a new task. Accordingly, this work proposed a collaborative relevance assessment approach to analyze the relevance of tasks and codified knowledge in the repository. And then, generating a task profile to support the proactive delivery of task-relevant knowledge. Note that the task profile specifies the key subject of an executing task. Table 1 shows the process of collaborative task-relevance assessment.

Once a knowledge worker is received or assigned a task, he/she will conduct the task-relevance assessment to generate the task profile. The proposed collaborative task-relevance assessment approach is a two-phase process. The *task categorization database,* which records the relevance degrees of existing tasks to categories, is used to support the operation of proposed two-phase task-assessment approach (as mentioned in Chapter 4). The worker conducts category assessment (phase-1 assessment) to derive the relevance degrees of the executing task (target task at hand) to categories. The system then selects a set of referring tasks (relevant or irrelevant) from existing tasks based on the similarity measures between the relevance degrees of tasks to categories. Consequently, the knowledge worker can conduct further task assessment (phase-2 assessment) without reviewing all existing tasks. Assessing the degree of relevance of tasks and categories is obtained by using the fuzzy linguistic approach. The details are described in Section 5.3.

Furthermore, a collaborative relevance feedback (RF) technique modified from the standard RF technique is adopted in this work to construct and refine the task profile of the executing task. The standard Rocchio and Ide_Dec_Hi in standard RF technique methods is modified by considering the relevance degrees of referring tasks obtained from fuzzy linguistic assessment. The modification considers the relative importance of relevant (positive) and irrelevant (negative) tasks from the perspective of users. Thus, a task-based knowledge support system can be realized with the proposed systematic profile modeling approach. The generated task profile is the system kernel that streamlines knowledge retrieval activity to further realize task-based knowledge support.

Table 1. Process of collaborative task-relevance assessment

```
Procedure   The collaborative relevance-assessment approach
  /* The following scenario detailed how the proposed concepts help the knowledge
     worker reuse historically tasks and then solve the encounter problem
     effectively. */
begin
  1. Assigned or choosing a task: The knowledge worker is assigned or
     received a task from set of executing tasks within the department.
       For example: "JayLee" is the executor of  "Recommendation in
     Composite e-Service" task.

  /* Entering the task assessment procedure to generate the task profile */
  2. Phase-1 assessment:
     2.1. The knowledge worker assesses relevant categories.
        Category set={ category1: Business Intelligence & Knowledge
                       Management, category2: Data Warehousing & Data
                       Mining, category3: IS Security, category4: Workflow
                       & e-Service and, category5: Internet Commerce}.
        The knowledge worker conduct relevance assessment by linguistic
      ratings;
    2.2 Assessing relevant categories collaboratively
        System operation: Load task-relevant experts or workers;
```

"*Kevin*" and "*Jack*" are the experts of "*Recommendation in Composite e-Service*" task

(As the system snapshot in Appendix A.)

Workers loads the relevance ratings of task-relevant experts;

System operation: Aggregating the relevance rating of all evaluators (the executor and task-relevant experts);

System operation: Submitting the aggregating vector and selecting the referring tasks from the existing tasks set;

/* The referring task set is the top most similar tasks in the existing task set */

System operation: Presenting the referring tasks in the system interface;

3. Phase-2 assessment

3.1. The knowledge worker assesses relevant tasks from the referring task set;

The knowledge worker conduct relevance assessment by linguistic ratings;

"JayLee" rated the following past task as the relevant task to the executing task:

<Existing-task 5, "*Designing Composite e-Services Platform with Recommendation Capability*", high>.

3.2 Assessing relevant tasks collaboratively

Workers loads the relevance ratings of task-relevant experts;

"Kevin" rated the existing-task 9 as the relevant task:

<Existing-task 9, "*Towards a Framework for Discovering Project-Based Knowledge Maps*", *very high*>.

System operation: Aggregating the relevance rating of all evaluators (the executor and task-relevant experts);

System operation: Submitting the assessment results to the system;

/*Obtaining the required task-relevant knowledge based on the construct task profile*/

4. Task-based Knowledge support

System operation: Constructing task profile of the executing task based on the assessment result.

/* Based on B-RA or F-RA methods as Equation (5.?) or Equation (5.?) */

System operation: Presenting and organizing the retrieval result in the system interface (As the system snapshot in Appendix B)

Knowledge support (relevant tasks and peer-groups): top-N task-relevant tasks in the task corpus, and the associated knowledge

workers in the relevant tasks; **Knowledge support (relevant documents and keywords):** top-N task-relevant documents in the task repository and top-N keywords in the task profile; ***Recommendation in Composite e-Service:*** ("e-service", 0.86), ("composite", 0.45), ("topic", 0.34), ("map", 0.34), ("groupware-based", 0.27), ("service", 0.23), ("knowledge", 0.23), ("k-discovery", 0.19), ("platform", 0.18), ("recommend", 0.14)…

＊Note: System operation means the back-end operation of the system.

5.3 Collaborative task-relevance assessment

A collaborative task-relevance assessment is employed to generate task profiles based on the task corpora of existing tasks. The task profile of an executing-task (or a new task) is initially derived via analyzing the task contents, or alternatively using the corresponding task corpus. A collaborative relevance assessment with fuzzy linguistic evaluation is then used to discover a set of referring tasks that are relevant or irrelevant to the executing task. That is, a two-phase assessment is presented to systematically model the procedure of relevance assessment via the collaboration of cooperative workers. The details are described below.

5.3.1 Phase 1-Identifying referring tasks based on category assessment

Phase 1 of the assessment determines the relevance degrees of the executing task to categories. The referring tasks are then identified by calculating the similarity measures based on the relevance degrees of tasks to categories.

Step 1 of phase 1: Determine the semantic term set and corresponding fuzzy number

For modeling the workers' perceptions on *Relevance*, the system defines six linguistic terms from "*very low*" to "*perfect*" to represent different relevance degrees. Each worker has his/her own perception of the approximate value (fuzzy scale) for each linguistic term. The fuzzy scale of a linguistic term is often modeled as a triangular fuzzy number. The linguistic terms are displayed in the front-end interface to provide knowledge workers a more natural and easier way of relevance assessment, while the corresponding fuzzy number is in the back-end for the system to facilitate numerical computation of relevance ratings. Notably, evaluators may not have identical fuzzy numbers on six linguistic terms of "*Relevance*" owing to different perceptions of the linguistic terms. Table 2 lists six linguistic scales of corresponding fuzzy numbers determined by four evaluators. For example, the evaluator E_1's perception on the linguistic term "*very high*" is within the fuzzy scale of (0.6, 0.7, 0.8). The evaluator E_2's perception on the linguistic term "*very high*" is within the fuzzy scale of (0.6, 0.75, 0.9). Each evaluator can use the front-end interface to easily setup his/her own fuzzy number of each linguistic term, or simply use the default fuzzy number provided by the system

A linguistic variable, *Relevance,* is defined to represent the degree of relevance between items (tasks or categories) assessed by evaluators. *E(Relevance)* is characterized using a fuzzy set of a universe of discourse *U*=[0,1], in which six linguistic terms $\check{r}_j$ and their associative semantic meanings $m(\check{r}_j)$ are defined as follows.

E(Relevance) = { $\check{r}_0$ = very low, $\check{r}_1$ = low, $\check{r}_2$ = normal, $\check{r}_3$ = high, $\check{r}_4$ = very high, $\check{r}_5$ = perfect}

where $m(\check{r}_i) < m(\check{r}_j)$, for $i < j$, and all $m(\check{r}_j)$ are distributed in [0,1].

The anti-symmetric distributed term set [34] is adopted, where more positive linguistic terms are defined, as shown in the defined term set, since this work places more emphasis on positive feedback to items. The fuzzy linguistic approach models the meaning of each term by fuzzy numbers. This work employs triangular fuzzy number (TFN), as defined in *Definition II* of Appendix A-1, to express the fuzzy scale of each linguistic term. TFN is widely used owing to its simplicity and solid theoretical basis [59], and thus is used to represent each linguistic term of the "*Relevance*" variable.

Step 2 of phase 1: Assess the relevance of task to categories collaboratively

This step mainly assesses the relevance of the executing-task to each category. The executor, namely the knowledge worker with the executing-task at hand, rates the relevance of executing-tasks to each category by linguistic terms (e.g. low, high etc.). Linguistic ratings denote the rating given in linguistic terms for the remainder of this paper. In addition, task-relevant experts or colleagues can also rate the relevance of executing-tasks to each category by linguistic terms to achieve collaborative

Table 2. Corresponding fuzzy numbers of linguistic term set by different evaluators

Evaluators	**VL** (Very Low)	**L** (Low)	**N** (Normal)	**H** (High)	**VH** (Very High)	**P** (Perfect)
E_1	(0,0.2,0.4)	(0.3,0.4,0.5)	(0.4,0.5,0.6)	(0.5,0.6,0.7)	(0.6,0.7,0.8)	(0.7,0.8,0.9)
E_2	(0,0.1,0.2)	(0.1,0.3,0.5)	(0.4,0.5,0.6)	(0.5,0.6,0.7)	(0.6,0.75,0.9)	(0.7,0.9,1)
E_3	(0,0,0)	(0.1,0.25,0.4)	(0.3,0.4,0.5)	(0.6,0.7,0.8)	(0.7,0.8,0.9)	(0.8,0.9,1)
E_4	(0,0,0)	(0.1,0.3,0.5)	(0.5,0.6,0.7)	(0.6,0.7,0.8)	(0.7,0.8,0.9)	(0.8,0.9,1)

assessment. Collaborative assessment, where a comprised rating is derived by aggregating ratings of task-relevant experts or colleagues is especially useful for the executor who is unfamiliar with the executing task. The linguistic ratings cannot be used by the system to calculate aggregate ratings, and thus need to be transformed into crisp ratings. In the front-end, linguistic terms are used. In the back-end, the system transforms the linguistic ratings into crisp ratings. Four evaluators determine the degree of relevance of the executing task t_e to each category using linguistic ratings based on their subjective judgments, as listed in Table 3(A). The corresponding fuzzy number of each linguistic rating is transformed into crisp numbers (ratings). For example, evaluator E_1's perception of the linguistic term "very high" is within the fuzzy scale of (0.6, 0.7, 0.8). The fuzzy number is transformed into a crisp value, 0.7, as shown in Table 3(B).

The fuzzy linguistic approach models the meaning of each term using fuzzy numbers. To achieve computational advantage, the crisp ratings (Best Non-fuzzy Performance values; BNP) are extracted from fuzzy numbers. Various methods can be used to defuzzify fuzzy numbers, including mean of maximal (MOM), center of area (COA), bisector of area (BOA), and so on [38].

This work adopts the COA method to calculate fuzzy numbers, owing to its simplicity and practicability. The COA method calculates the fuzzy mean under uniform probability distribution assumption [45]. If the fuzzy number $\tilde{Z}$ is triangular, the crisp rating can be derived by the equation: $CV(\tilde{Z}) = [(r-l)+(m-l)]/3+l$. For example, Table 3(B) lists the crisp ratings transformed from the linguistic ratings of the evaluators based on the above equation.

Step 3 of phase 1: Aggregate the relevance ratings of evaluators

Evaluators' crisp ratings obtained from collaborative assessment are aggregated in this step. The relevance degree of the executing task to each category is derived by computing the weighted average of evaluators' crisp ratings of

Table 3. Assess the relevance of executing task to categories

(A) Assessment by linguistic terms

Category	**Evaluator**			
	E_1	**E_2**	**E_3**	**E_4**
C_1	N	N	N	H
C_2	VH	H	N	VH
C_3	P	VH	VH	H
C_4	VL	N	L	L

(B) Crisp ratings derived from linguistic ratings

Category	**Evaluator**			
	E_1	**E_2**	**E_3**	**E_4**
C_1	0.5	0.5	0.4	0.7
C_2	0.7	0.6	0.4	0.8
C_3	0.8	0.75	0.8	0.7
C_4	0.2	0.5	0.25	0.3

relevance to the category. The aggregated relevance of the executing task to categories is expressed as a vector of relevance degree to each category. The top-N similar tasks and last-M non-similar tasks can then be retrieved based on the similarity measures between the relevance degrees of tasks to categories, as detailed in Step 4. Let $A_{ej}(c_i)$ denote the crisp rating of evaluator e_j regarding the relevance of the executing task t_e to category c_i. Moreover, let w_{ej} denote the associated weight representing the relative importance (weight) of the rating of evaluator e_j. The aggregated relevance of the executing task to category c_i, $A_E(c_i)$, is derived as $\sum_j w_{ej}A_{ej}(c_i)$. Notably, the relevance degree of task t_e to categories can be modeled as a vector $\overrightarrow{t_e^C}$.

$$\overrightarrow{t_e^C} =< A_E(c_1), A_E(c_2), ..., A_E(c_m) >$$

Continuing from the assessment listed in Table 3, if $w_{ej} = 1/n_e$, where n_e denotes the number of evaluators, then the aggregated relevance ratings are calculated as the arithmetic mean. The aggregated relevance degrees of the executing task to categories are expressed as follows.

$$\overrightarrow{t_e^C} =< A_E(c_1), A_E(c_2), A_E(c_3), A_E(c_4) >=< 0.525, 0.625, 0.7625, 0.3125 >$$

Step 4 of phase 1: Select referring tasks

The proposed mechanism reduces the number of tasks to be assessed via discovering a set of referring tasks to assist workers in conducting task-relevance assessment (phase 2 assessment). This step identifies a subset of existing tasks as referring tasks based on their similarity to the executing task derived using the relevance degrees of tasks to categories. Notably, relevance degrees of the executing task to categories are derived by Step 3 of the category assessment. A similarity (cosine) measure is adopted to calculate the similarity between the executing task and an existing task based on their relevance degrees to categories. Based on the similarity measures, the top-N similar tasks are chosen as the positive (relevant) referring tasks, whereas the last-M non-similar tasks are chosen as the negative (irrelevant) referring tasks. The referring tasks are used for further task-relevance assessment in phase 2.

The similarity measure between the executing task t_e and an existing task t_r can be computed as the cosine of the angle between two vectors, $\overrightarrow{t_e^C}$ and $\overrightarrow{t_r^C}$, namely *cosine*($\overrightarrow{t_e^C}$, $\overrightarrow{t_r^C}$). Notably, $\overrightarrow{t_e^C}$ is derived by the collaborative relevance assessment as described in Step 3, while $\overline{t_r^C}$ is derived by the fuzzy classification, as described in Chapter 4.

Example: The aggregated relevance degrees of the executing task t_e to categories is modeled as a vector $\overline{t_e^C}$, $\overline{t_e^C} =< 0.525, 0.625, 0.7625, 0.3125 >$. Table 4 lists the relevance of ten existing tasks to categories. The similarly measures of executing task and existing tasks are listed in the sixth column of Table 4. The ranking of similarity measures is displayed in the last column. The top-5 tasks, t_3, t_4, t_5, t_9 and t_{10}, are selected as the positive referring tasks, while the last-2 tasks, t_2 and t_6, are chosen as the negative referring tasks.

5.3.2 Phase2-Assessing the relevance of referring tasks

Phase 2 conducts an assessment to determine the relevance of the referring tasks to the executing task. The evaluators assess the degree of relevance between the executing task and referring tasks without reviewing all tasks. The task assessment procedure resembles the procedure of category assessment. The evaluators conduct relevance assessment to determine the relevance degree of each referring task to the executing task. They use linguistic terms to rate the relevance of each referring task to the executing task. The aggregated relevance rating of a referring task is derived by computing the weighted average of evaluators' crisp ratings on the relevance of the referring task to the executing task. The relevance degrees of referring tasks to the executing task are then used to construct the task profile of the executing task detailed in Section 5.4.

Let $A_{ej}(t_r)$ represent the crisp rating of the evaluator e_j on the relevance of the executing-task to a referring task t_r. Moreover, let w_{ej} denote the associated weight

Table 4. Relevant degree between tasks and categories

	C1	C2	C3	C4	Similarity	Ranking
Task 1	0	0.25	0.22	0	0.838	(6)
Task 2	0	0	0	0.71	0.269	(10)
Task 3	0	0.25	0.24	0	0.844	(5)
Task 4	0.12	0.19	0.29	0	0.947	(3)
Task 5	0	0.11	0.13	0	0.850	(4)
Task 6	0	0	0.68	0	0.657	(9)
Task 7	0.49	0.11	0.15	0	0.724	(8)
Task 8	0	0.15	0.54	0	0.778	(7)
Task 9	0.18	0.18	0.29	0	0.957	(1)
Task 10	0.13	0.21	0.27	0	0.955	(2)

representing the relative importance (weight) of the rating of evaluator e_j. The aggregated relevance rating of task t_r to the executing-task, $A_E(t_r)$, is derived as $\sum_j w_{ej}A_{ej}(t_r)$.

Example: Table 5(A) lists the linguistic ratings on five positive referring tasks evaluated by four evaluators. Moreover, Table 5 (B) lists the crisp ratings of tasks 5 and 9, derived from the linguistic ratings. The aggregated relevance ratings, $A_E(t_5)$=0.675 and $A_E(t_9)$=0.8375, are also listed in the last column of Table 5 (B).

Table 5. Assessment on the relevance of positive referring tasks to the executing task

(A) Assessment by linguistic terms

Referring Task	**Evaluator**			
	E1	**E2**	**E3**	**E_4**
Task 3	VL	VL	H	VL
Task 4	VL	VL	VL	VL
Task 5	H	H	VH	H
Task 9	P	VH	P	P
Task 10	VL	H	H	H

(B) Crisp ratings derived from linguistic ratings

Referring Task	**Evaluator**				**Aggregated relevance rating of tasks $A_E(t_r)$**
	E1	**E2**	**E3**	**E_4**	
Task 5	0.6	0.6	0.8	0.7	**0.675**
Task 9	0.8	0.75	0.9	0.9	**0.8375**

5.3.3 Discussions

In this work, the task-related experts of each task are predefined. In addition, the relative importance of experts is given the same weight to aggregate the relevance ratings. That is, the aggregated relevance ratings are calculated as the arithmetic mean. In the future, we shall consider revising our group decision method with the aid of recommendation techniques in Recommender system. Accordingly,

- We will employ methods e.g., collaborative filtering algorithm, demographic profiles of workers, etc, to determine task-related experts. Thus, the new-system cold-start problem may encounter by the demographic profiles of workers and the new-user cold-start problem my encounter by the hybrid recommendation technique.
- Furthermore, the relative importance of task-related experts could be determined by the calculation result of recommendation algorithms.

5.4 Task-based $\mathcal{K}$-Support based on assessment

5.4.1 Task profile generation by relevance feedback technique

The task profile of an executing task is specifies key subjects of the executing task, which represented as a feature vector of weighted terms. The task profile is initially derived via analyzing the task contents, or alternatively using the corresponding task corpus. Moreover, the collaborative task-assessment identifies the relevance degrees of referring tasks to the executing task. The result is used to further construct or refine the task profile of the executing task based on the relevance feedback (RF) techniques introduced in Chapter 2.

The RF technique employs the process of reformulating or expanding the original query based on partial relevance judgments, i.e. feedbacks on part of the evaluation set. The RF technique is adopted in this work to construct and refine the task profile of the executing task. Two kinds of relevance judgment on referring tasks are considered: positive feedback and negative feedback. The standard RF technique employs binary feedback without considering the degrees of relevance, as shown in Eq. 5.1. Relevant tasks with positive feedback give positive influence on the weights of terms occurring in relevant tasks. The irrelevant tasks with negative feedback give negative influence on the weights of terms occurring in irrelevant tasks. A refined task profile can be generated by adding the term weights of relevant tasks and subtracting the term weights of irrelevant tasks. Consequently, the feature vector of new term weights derived based on the RF technique forms a new task profile for further knowledge retrieval. The idea of relevance feedback shifts the new profile closer to the relevant task set and away from the irrelevant task set. The parameters β and γ are used to determine the relative amount of influence of the relevant task set to the irrelevant task set.

This work modifies the standard Rocchio and Ide_Dec_Hi methods by considering the relevance degrees of referring tasks obtained from fuzzy linguistic assessment. The modification considers the relative importance of relevant (positive) and irrelevant (negative) tasks from the perspective of users. The feature vectors of referring tasks are multiplied with their relevance degrees to reflect their relative contributions in the refinement of the task profile, as expressed in Eq. 5.2. Detailed formulations are described as follows.

Two approaches are proposed for constructing the task profile $\vec{S}_e$ of executing task t_e based on the relevance assessment introduced in the previous section. The binary relevance assessment method, denoted as B-RA, conducts binary (relevant and irrelevant) assessment. The fuzzy linguistic relevance assessment method, denoted as F-RA, considers the relevance degree based on user perceptions.

$$\text{B-RA: } \vec{S}_e = \alpha \vec{S}_{initial} + \beta \sum_{\forall t_j \in T_r} \vec{t}_j - \gamma \sum_{\forall t_j \in T_n} \vec{t}_j \tag{5.1}$$

$$\text{F-RA: } \vec{S}_e = \alpha \vec{S}_{initial} + \beta \sum_{\forall t_j \in T_r} (w_{t_j}) \vec{t}_j - \gamma \sum_{\forall t_j \in T_n} (1 - w_{t_j}) \vec{t}_j \tag{5.2}$$

Where $\vec{S}_{initial}$ represents the initial profile derived from analyzing the collected relevant documents for the executing task, if available. Moreover, T_r denotes the set of relevant tasks selected from positive referring tasks according to collaborative assessment of experts and workers. T_n represents the set of the last-M irrelevant tasks that are selected by the system automatically (t_2 and t_6, in the previous given example, Table 3). Furthermore, $\vec{t}_j$ is the task corpus of task t_j with an associated weight w_{tj} representing the relevance degree of t_j to the executing task. w_{tj} is set to $A_E(t_j)$, which is the aggregated relevance rating of task t_j to the executing task. $A_E(t_j)$ is derived from the task assessment procedure illustrated in Section 5.1.2, and α, β and γ are tuning constants.

The task profile of the executing task t_e, derived from Eq. 5.1 or 5.2 can be expressed as a feature vector of weighted terms, $\vec{S}_e = \langle w(k_1, t_e), w(k_2, t_e), \ldots, w(k_n, t_e) \rangle$, where $w(k_i, t_e)$ is the weight of a term k_i in representing the main subjects of t_e; n denotes the number of discriminating terms. Meanwhile, $\vec{S}_e$ is used to retrieve relevant codified knowledge from the repository.

5.4.2 K-Support: task-based knowledge retrieval

A task-based knowledge support system can be realized with the proposed systematic profile modeling approach. The generated task profile is the system kernel that streamlines knowledge retrieval activity for further realizing task-based knowledge support. A task profile specifies key subjects of the executing task, and is constructed to model the information needs of knowledge workers during task execution. Based on task profiles, the system can recommend/retrieve relevant knowledge from the repository to assist knowledge workers. Workers conduct further search activity are assisted by the highly correlated term set presented in the system interface. The relevant knowledge includes relevant tasks, associated peer groups, relevant documents, and highly correlated term set.

The similarity measures between the executing task and the codified knowledge items can be calculated to select Top-N relevant tasks or documents from the knowledge repository. The key contents of a codified knowledge item (task or document) are represented as a feature vector of weighted terms. The task profile is also expressed as a feature vector of weighted terms. The cosine measure of feature vectors described in Section 2.2.1 can be used to derive the similarity measure.

Moreover, the task profile can be further adjusted during task performance by monitoring the workers' feedback behavior. The most task-relevant codified knowledge can be retrieved based on the adjusted task profile to fit the worker's current information needs. We also presented an adaptive task-based profiling approach to model workers' dynamic task needs. The task-based peer-groups can be analyzed from the retrieved relevant task set to provide knowledge sharing. Details for identifying task-based peer groups to support knowledge sharing are given in next chapter and can be found in our recent work [47].

Relevant Tasks Recommendation. As the task profile $\vec{S}_e$ (feature vector) of the executing task t_e has been derived by B-RA or F-RA method, retrieving relevant tasks for references will be helpful. The cosine measure of $\vec{S}_e$ and $\vec{t}_j$, namely, *cosine*$(\vec{S}_e, \vec{t}_j)$, is calculated as the similarity measure between the executing task and task t_j. Notably, $\vec{S}_e$ and $\vec{t}_j$ are the feature vectors of t_e and t_j, respectively. Tasks with top-N similarity measures are selected as the relevant tasks for recommendation. The relevant tasks and associated knowledge workers engaged in these relevant tasks are recommended for consultation. Effectively codifying tacit knowledge may be difficult. However, the system can locate valuable knowledge sources such as knowledge workers engaged in relevant tasks, providing a knowledge support platform for gathering and exchanging task-relevant knowledge among workers.

Relevant Documents and Term Recommendation. The relevant documents are retrieved using the profile of the executing-task. Similarity measurement is also adopted to select top-*N* relevant documents. Let $\vec{t}_j$ are the feature vector of document d_j. The cosine measure of $\vec{S}_e$ and $\vec{d}_j$, namely, *cosine*$(\vec{S}_e, \vec{d}_j)$, is calculated as the similarity measure between the executing task and document d_j. Documents with top-N similarity measures are selected as the relevant documents for recommendation. Meanwhile, the important term set representing the main subjects of the executing task is derived from the constructed task profile $\vec{S}_e$. The system displays the discriminating terms and their associated weights to assist knowledge workers with further retrieval. The term set forms the task corpus of the executing task, and can be modified during the subsequent stages of task execution.

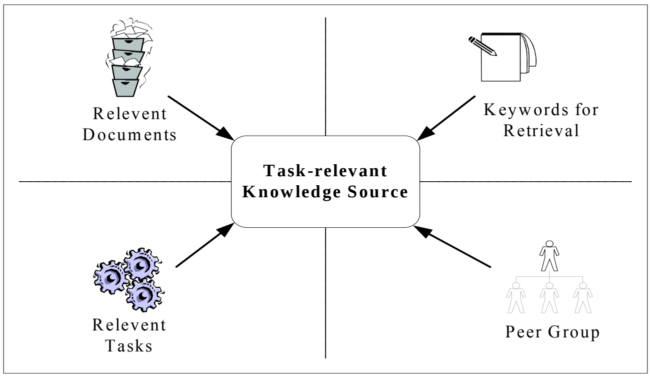


Fig. 4. Task-relevant knowledge source (explicit or tacit knowledge source)

5.5 Experimental setup

Three experiments were performed to evaluate the assessment and retrieval effectiveness based on the proposed methods. Section 5.5.1 review the experiments and the experimental procedure, respectively. Meanwhile, the remaining subsections describe the participants, evaluation metrics and related parameter selection.

5.5.1 Overview of experiments

Experimental objective and design

Three experiments are conducted to evaluate the effectiveness of the proposed *collaborative relevance-assessment approach*. The objectives of experimental evaluations were threefold: (1) Experiment one evaluates if building task profiles based on binary or fuzzy linguistic relevance assessment method can help knowledge workers retrieve task-relevant information more precisely than the query-based method without profile generation; the experiment also evaluates the effectiveness of fuzzy linguistic assessment for two user groups: experienced users and novices; (2) Experiment two evaluates if the proposed two-phase relevance assessment approach (denoted as 2-F-RA) can ease the assessment load on a large number of tasks; and (3) Experiment three evaluates if the proposed collaborative relevance-assessment can help knowledge workers find task-relevant information more precisely by the aid of domain experts.

Experiment one compares the binary relevance assessment method (B-RA method) and the fuzzy linguistic relevance assessment method (F-RA method) with the Query-based method. The B-RA and F-RA methods are one-phase relevance assessment methods that only conduct task-relevance assessment (phase 2), as described in Section 5.3.2, without employing phase-1 category-relevance assessment and collaborative assessment. The query-based method simply employs traditional keyword search supported in a search engine to access knowledge items without profile generation. The method provides a user-driven approach for knowledge workers to express their information needs as queries to search for needed knowledge items. Experiment two measures the impact of assessment load while conducting task-relevance assessment. Accordingly, the two-phase relevance assessment approach (denoted as 2-F-RA) is compared with the one-phase relevance assessment approach, F-RA. The 2-F-RA method conducts both the phase-1 category relevance assessment (Section 5.3.1) and the phase 2 task-relevance assessment (Section 5.3.2) without employing collaborative assessment. Notably, the phase-1 assessment determines the relevance of the executing task to categories, and then identifies the referring tasks by computing the similarity measures based on the relevance degrees of tasks to categories. The third experiment evaluates the effectiveness of collaborative two-phase assessment (denoted as Collaborative 2-F-RA) versus non-collaborative two-phase assessment (denoted as 2-F-RA). The collaborative assessment aggregates the relevance ratings of evaluators derived from the assessments of experts and collaborative workers.

Experimental procedure

Figure 5 shows the overall experimental procedure. Solid arrows indicate the experiment flow, while dashed arrows indicate the interaction between the system and the database. A set of referring tasks are selected from the category assessment result with the support of category schema (denoted by circle 1) such that the participant can conduct task assessment to construct a task profile based on the referring tasks (denoted by circle 2). The participant may adopt collaborative assessment (denoted by circle 3), where the relevance ratings of experts are combined with the relevance ratings of participants to obtain final aggregated relevance ratings, as described in Section 5.3. The relevance ratings are incorporated into the relevance feedback to generate task profiles, as presented in Section 5.4. The system then retrieves task-relevant knowledge items from the task-oriented information repository via computing and ranking the similarity measures between the task profile and codified knowledge (denoted by circle 4). The effectiveness is evaluated in terms of recall to precision curve.

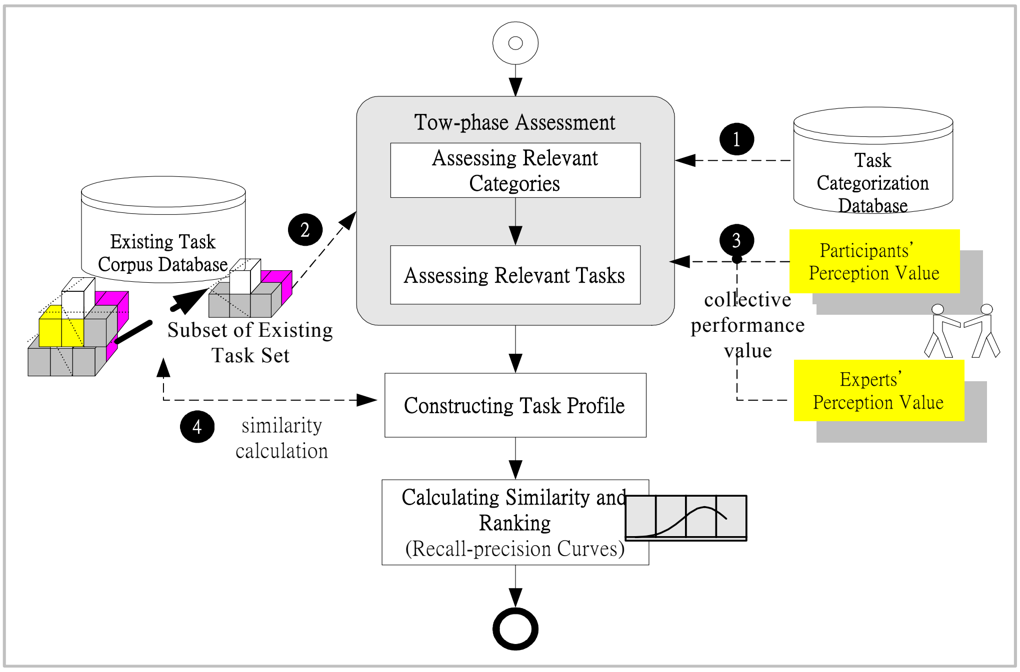


Fig. 5. Experimental procedure

5.5.2 Data, participants and evaluation metrics

Data

Experiments using a real application domain were carried out for conducting research tasks in a research institute laboratory. The tasks constitute writing research papers or conducting research projects. The real application domain restricts the sample size of the data and participants in the experiments.

Fifty research tasks were collected, with 31 existing tasks and 19 executing tasks. Over 250 documents accessed by tasks are collected during the period of 2002 ~ 2003. The smallest meaningful components of document information elements, such as title, abstract, journal and author, were extracted from documents. Each document contained an average of 90 distinct terms after information extraction, and document pre-processing (e.g. case folding, stemming, and stop word removal).

An existing task is the historical tasks that had been accomplished in the research institute. The task corpus of each existing task, namely a feature vector of weighted terms, is derived by analyzing the documents generated and accessed by the existing task, also described in Section 4.1. Fuzzy classification is performed to derive the relevance degrees of existing tasks to categories as described in Section 4.2. Existing tasks are classified into five categories. The *task categorization database* records the relevance degrees of each existing task to categories. On the other hand, an *executing task* is the target task that the knowledge worker conducts at hand. The task profile of an executing-task (or an on-going task) is derived based on the task corpora of existing tasks and their relevance to the executing task as described in Section 5.3.

Participants

Knowledge workers usually require a longer time (eg. one year) to accomplish knowledge-intensive tasks. However, it is difficult to design experiments relevant to real world problems, when the task performance process spans a long time period. Thus, we choose evaluators according to their task execution progress, classified into two levels, familiar or unfamiliar with the executing tasks. Consequently, two user groups were chosen for conducting the experiments: experienced workers familiar with the executing task, and novices unfamiliar with the executing task. The number of experimental participants is also restricted.

Six executing tasks were chosen as the testing set for evaluations, as lists in Table 6. And eighteen workers are selected to participate in the experiments. Note that two participants gave up the testing. To evaluate the effectiveness of collaborative relevance-assessment, executing tasks in the testing set are those with more than one

Table 6. Six selected executing tasks (on-going tasks)

Task	Task Name	Task Characteristic
1	A Study of Feature-Weighting Clustering in Recommender	Proposal of Thesis
2	Comparisons of Collaborative Filtering for	Proposal of Thesis
3	News Detection and Tracking based on Event Hierarchy	Proposal of Thesis
4	Deployment of Composite e-Service Framework	System Development
5	Recommendation in Composite e-Service	Research project
6	Modeling of Process-View based Workflow Management	Research project

knowledge worker participating in the task. Moreover, we chose executing tasks conducted by at least one novice and one experienced worker to evaluate the effectiveness of proposed methods for different user groups. We randomly selected one or two experienced workers and one or two novices from each testing task as participants in the testing set. The testing set selection limitation for the problem domain also restricts testing set size.

Performance evaluation metrics

The retrieval effectiveness is plotted as a recall-precision curve, which treats precision as a function of recall [7][83].

Precision and recall. *Precision* is the fraction of retrieved items (tasks or documents) that are relevant, while *recall* is the fraction of total known relevant items that are retrieved, defined as Eq. 5.3 and 5.4.

$$precision = \frac{|retrieved\ items\ that\ are\ relevant|}{|total\ retrieved\ items|} \tag{5.3}$$

$$recall = \frac{|relevant\ items\ that\ are\ retrieved|}{|total\ known\ relevant\ items|} \tag{5.4}$$

Notably, both the number of total retrieved items and the number of total known relevant items must be greater than zero. Increasing the number of retrieved items tends to reduce precision and increase recall. Generally, precision is high at low recall levels and low at high recall levels. Thus, the recall-precision curve is used to show the interpolated precision at each recall level, as follows. The recall values can be divided into different recall levels with rv_i, $i \in \{1,2,\ldots,n\}$, denoting a reference point to the *i*-th recall level. The *interpolated* precision, $IP_r(rv_i)$ thus can be expressed as: $IP_r(rv_i)=MAX\ P_r(rv)\ for\ rv_i \leq rv < rv_{i+1}$, where $P_r(rv)$ represents the precision value given a recall value of rv.

The *interpolated precision* at each recall level can be derived for each task being evaluated. For evaluating a set of tasks, the average *interpolated precision* is derived as Eq. 5.5.

$$aveIP_r(rv_i) = \sum_{i=1}^{k} \frac{IP_r(rv_i)}{k} \tag{5.5}$$

where $aveIP(rv_i)$ denotes the average *interpolated precision* at the *i*th recall level, and k denotes the number of evaluated tasks.

Evaluation criteria

The effectiveness of task-based knowledge support method is measured by recall and precision. In this experiment, we want to evaluate the tradeoff between recall and precision and the system overall performance at each recall level. Thus, the relationship between precision and recall is depicted in terms of *recall-precision curve*. That is, we observe the trend of *recall-precision curve* (the curve decreased, increased or even horizontal lines) and the value of precision at each recall level to discuss the system search capability.

We don't just choose precision as the performance metric, since it is not easy to observe the overall system performance of various methods. And only shows the system performance under top-N document or task support. Even we choose F-Measure as the performance metrics, the metric only shows the tradeoff between recall and precision and cannot depict the system overall performance. In fact, a "perfect" search system, precision would be 100 percent at all recall level; therefore, the *recall-precision curve* is a horizontal line at 100 percent. However, the precision is generally high at low recall level and low at high recall levels, the trend of curve is decreasing. Thus, if one method's curve is completely above the other one, then the method is better than the other. Accordingly, we make comparisons between methods, including Query-based method (baseline method), B-RA, F-RA, 2-F-RA, Collaborative 2-F-RA method by *recall-precision curves*. And we will discuss the advantage and disadvantage of the methods from three aspects.

(1) The average precision value, $avgIP(rv_i)$, to verify the search performance between the methods.
(2) Looking more specifically, showing the value of precision at each recall level to examine the tradeoff between methods.
(3) Drawing the *recall-precision curves* to show the overall performance of methods.

A good condition is the method always has high precision at each recall level and far better than the other methods. However, some methods may have low average precision value, but have high precision at a certain interval of recall level.

5.5.3 Parameter selection

This work adopts and modifies the classical relevance feedback methods, standard

Rocchio and Ide_Dec_Hi methods described in Eq. 2.3 and 2.4 of Section 2.2.2, to design the proposed relevance feedback methods. The study of Salton and Buckley (1990) has suggested the steps of a pilot experiment to determine the parameters of the two classical relevance feedback methods. Their result suggests that setting α=1, β=0.75, and γ=0.25 can achieve better retrieval performance (higher precision value). Most studies suggest that the information of relevant documents is more important than that of irrelevant documents [32][68]. This work uses the similar approach suggested by Salton and Buckley to determine the parameter setting.

In this work, we conduct a pilot experiment to determine the parameter values of α, β and γ in the proposed equations described in Eq. 5.1 and 5.2. This work sets α = 1 and $\beta+\gamma$ = 1 to adjust the relative importance of relevant and irrelevant tasks. Accordingly, only one single parameter needs to be determined (β or γ). The experiment is conducted by systematically adjusting the value of β with the scale of 0.1. The *precision* (as shown in Eq. 5.3) metric is chosen as the performance measure to evaluate the effectiveness of the methods. The optimal parameter values with the best results (the highest precision value) are chosen as the parameter settings of the proposed equations. The experimental results suggested that best result can be achieved by setting α = 1, β = 0.8 and γ = 0.2. This finding agrees with the suggestion of most previous studies that the information of relevant documents is more important than that of irrelevant documents[32][66][68]. Subsequently, the parameters α = 1, β = 0.8 and γ = 0.2 are adopted in our experiments.

5.6 Experimental results and implications

5.6.1 Experiment one: effect on fuzzy linguistic assessment

The objective of this experiment is to evaluate the effectiveness of ***fuzzy linguistic assessment***. This experiment evaluates the effectiveness of finding task-relevant information via the query-based method and the methods that generate tasks profiles based on relevance assessment, binary relevance assessment (B-RA method) and the fuzzy linguistic relevance assessment (F-RA method), described in Section 5.3. The B-RA method employs binary (relevant and irrelevant) assessment and relevance feedback without considering the degree of relevance. The F-RA method considers the degree of relevance in the assessment and relevance feedback, namely, modeling the user's perception value by fuzzy linguistic rating approach. Moreover, the B-RA and F-RA methods conduct task-relevance assessment (phase 2) without employing phase-1 category-relevance assessment and collaborative assessment. These two methods are compared with the query-based method, the baseline method, to demonstrate that generating task profiles from existing tasks can help knowledge workers locate task relevant information more easily.

Table 7 shows the *average interpolated precision* at each recall level, computed over the evaluating tasks, for the three methods and two user groups. The recall level is represented as [rv_i, rv_{i+1}), denoting the interval of recall values that satisfy $rv_i \leq recall < rv_{i+1}$.The last row shows the average precision values computed over all recall levels.

Observation 1: The average precision values of both the B-RA and F-RA methods exceed those of the query-based method, for the experienced user and novice groups, respectively. The experimental result reveals that building task profiles by assessing the relevance to existing tasks can help knowledge workers retrieve task-relevant information.

Observation 2: For experienced knowledge workers, the average precision of F-RA is higher than that of B-RA. This analytical result indicates that the F-RA method can provide better knowledge support to experienced users than does the B-RA method.

Observation 3: Interestingly, for novices, the average precision value of B-RA closely approximates that of F-RA. Table 8 lists the detailed average precision values of task retrieval for ten individual novices. For three cases, the average precision value of F-RA is lower than that of B-RA. Moreover, some novices cannot obtain better knowledge support from F-RA than from B-RA. This analytical result implies

Table 7. Result of knowledge support for task retrieval (B-RA vs.F-RA)

	Experience Users			**Novices**		
Recall Level	**Query**	**B-RA**	**F-RA**	**Query**	**B-RA**	**F-RA**
	Precision	Precision	Precision	Precision	Precision	Precision
[0.0, 0.2)	1.000	1.000	1.000	1.000	1.000	1.000
[0.2, 0.4)	0.745	0.945	0.944	0.670	0.762	0.762
[0.4, 0.6)	0.645	0.833	0.889	0.566	0.648	0.644
[0.6, 0.8)	0.502	0.733	0.820	0.458	0.610	0.600
[0.8, 1.0)	0.497	0.616	0.623	0.403	0.402	0.407
[1.0, 1.0]	0.333	0.395	0.396	0.359	0.331	0.351
6-pt Average Precision	0.620	0.754	**0.779**	0.576	0.626	**0.627**

Table 8. Result of knowledge support for task retrieval by ten novices

	$N_1(T_1)$	$N_2(T_2)$	$N_3T_{(2)}$	$N_4(T_3)$	$N_5(T_4)$	$N_6(T_4)$	$N_7(T_5)$	$N_8(T_5)$	$N_9(T_6)$	$N_{10}(T_6)$
B-RA	0.778	0.549	**0.643**	**0.691**	**0.814**	**0.577**	0.450	0.644	**0.571**	0.462
F-RA	**0.786**	**0.559**	**0.643**	**0.691**	0.805	0.552	**0.467**	**0.647**	0.508	**0.475**

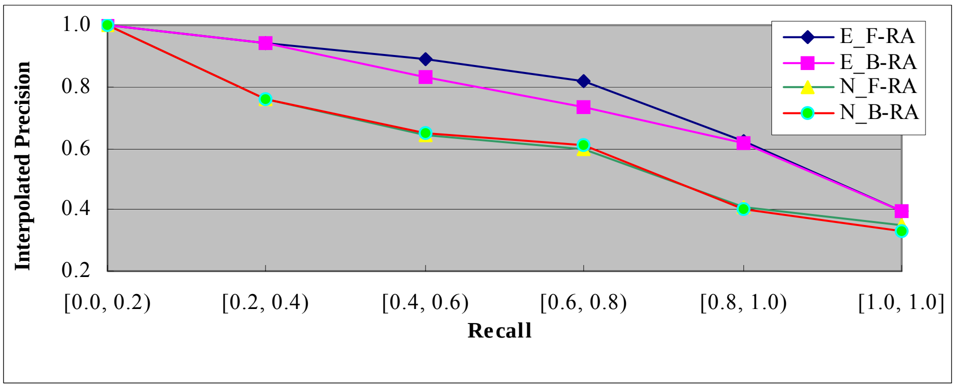


Fig. 6. Average recall-precision curves for experienced users and novices (B-RA vs. F-RA)

that experienced users are knowledgeable in making appropriate assessments regarding the degree of relevance using the fuzzy linguistic approach. However, some novices are not sufficiently knowledgeable to distinguish the relevance or irrelevance of the executing task and existing tasks. A simple binary assessment (relevant or irrelevant) may be more appropriate for novices when they are unfamiliar with the task.

Observation 4: Figure 6 plots the average recall-precision curves of the three proposed methods, and shows the gradual decrease in average precision value. Notably, **E_**(F-RA,B-RA) denotes "Experienced users" and **N_**(F-RA, B-RA) denotes "Novices". The experimental result reveals that both the average precision values of F-RA and B-RA for experienced users exceed those of F-RA and B-RA for novices. The proposed assessment also provides experienced users with better knowledge support than novices. Experienced users have more working experience, and thus are more knowledgeable to make proper relevance assessment.

Implications: The system-driven approach for proactive delivery of task-relevant knowledge by building task profiles can provide more effective knowledge support than user-driven Query-based approach to access knowledge items. Moreover, the result reveals that experienced users have more working experience, and thus are more knowledgeable than novices to make proper relevance assessment. The result implies that novices may gain benefit from the collaboration of experienced workers in conducting *task relevance-assessment*.

5.6.2 Experiment two: effect on two-phase relevance assessment

This experiment aims to evaluate whether reducing the number of referring tasks for assessment can assist users conduct task relevance assessment. This experiment evaluates the effectiveness of knowledge support by two-phase fuzzy linguistic relevance assessment (2-F-RA) and one-phase fuzzy linguistic relevance assessment (F-RA).

Conducting relevance assessment on a large number of tasks may place a burden on users and influence the assessment result. To measure the impact of assessment load during task-relevance assessment, this work compares the two-phase linguistic relevance assessment approach (2-F-RA) with the one-phase linguistic relevance F-RA approach. Notably, the experiment one has demonstrated that the average precision values of F-RA method exceed those of the B-RA and query-based methods, especially for the experienced users. Therefore, we chose F-RA method instead of B-RA method or query-based method in comparison with the 2-F-RA method. The 2-F-RA approach reduces the number of tasks by selecting referring tasks based on the category assessment (phase 1). The one-phase F-RA approach conducts task relevance assessment (phase 2) without performing phase-1 assessment.

Tables 9 and 10 show the effectiveness of task-retrieval and document-retrieval, using the F-RA and 2-F-RA methods, respectively. Moreover, Figure 7 plots the average recall-precision curves of two assessment methods based on Table 9.

Observation 1: The result shows that the overall average precision using the 2-F-RA method is higher than that of F-RA method for both user groups. The experimental result implies that the two-phase relevance assessment (2-F-RA) provides better knowledge support for task and document retrieval than the F-RA method. With the support of category assessment (phase 1) to reduce the burden of relevance assessment on a large number of tasks, the two-phase assessment can assist workers in conducting task-relevance assessment more effectively than the one-phase assessment.

Observation 2: Figure 7 shows that the effectiveness of knowledge support for experienced users exceeds that of novices. Experienced users (E_2-F-RA, E_F-RA) are more knowledgeable and thus can gain more effective knowledge support than novices (N_2-F-RA, N_F-RA) do.

Implications: The result reveals that the two-phase assessment can assist workers in conducting task-relevance assessment more effectively than the one-phase assessment. Category assessment (phase 1) is helpful to reduce the burden of

Table 9. Knowledge support for task-retrieval (2-F-RA versus F-RA)

Recall Level	Experience Users		Novices	
	F-RA	**2-F-RA**	**F-RA**	**2-F-RA**
	Precision	Precision	Precision	Precision
[0.0, 0.2)	1.000	1.000	1.000	1.000
[0.2, 0.4)	0.944	0.958	0.762	0.812
[0.4, 0.6)	0.889	0.945	0.644	0.728
[0.6, 0.8)	0.820	0.883	0.600	0.556
[0.8, 1.0)	0.623	0.659	0.407	0.461
[1.0, 1.0]	0.396	0.484	0.351	0.357
6-pt Average Precision	0.779	**0.822**	0.627	**0.652**

Table 10. Knowledge support for document retrieval (2-F-RA versus F-RA)

Recall Level	Experience Users			
	F-RA	**2-F-RA**	**F-RA**	**2-F-RA**
	Precision	Precision	Precision	Precision
[0.0, 0.2)	0.650	0.803	0.703	0.800
[0.2, 0.4)	0.306	0.351	0.203	0.226
[0.4, 0.6)	0.271	0.320	0.189	0.209
[0.6, 0.8)	0.227	0.241	0.168	0.174
[0.8, 1.0)	0.168	0.177	0.152	0.149
[1.0, 1.0]	0.144	0.145	0.134	0.137
6-pt Average Precision	0.294	**0.340**	0.258	**0.282**

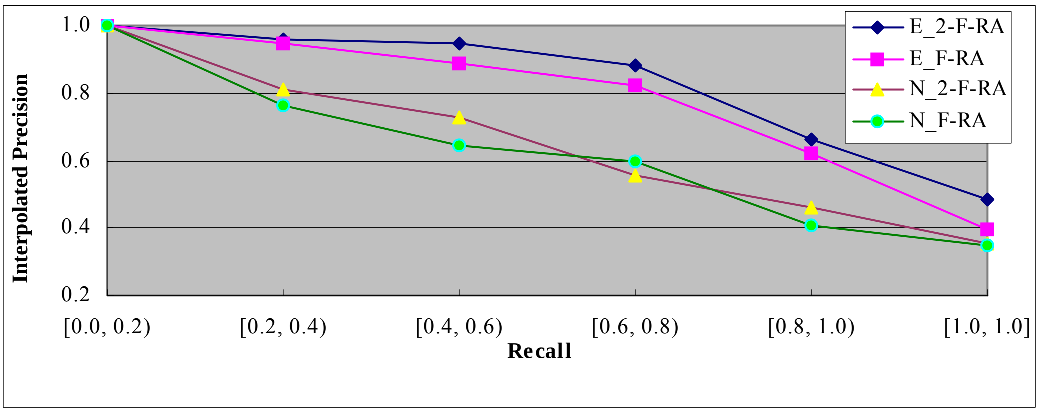


Fig. 7. Average recall-precision curves for experienced users and novices (F-RA vs. 2-F-RA)

relevance assessment on a large number of tasks. The result implies that reducing the number of items for assessment can decrease knowledge worker's burden and cognitive load in conducting relevance assessment. Consequently, more effective knowledge support can be achieved.

5.6.3 Experiment three: effect on collaborative assessment

The objective of this experiment is to show that collaborative assessment can reduce the workload of novices and help them find task-relevant information. Novices who are less knowledgeable about and unfamiliar with tasks in the initial working stage may have difficulty in performing task-relevance assessment. Several studies on knowledge management have demonstrated that expert opinion is crucial source of knowledge support for novices. This work employs collaborative assessment to help novices with the aid of experts and collaborative workers. The effect of collaborative assessment is also compared with that of non-collaborative assessment.

The result of experimental two showed that the two-phase relevance assessment (2-F-RA) provides better knowledge support for task and document retrieval than the F-RA method. This experiment takes a further step to evaluate the effectiveness of collaborative relevance assessment by experienced users and novices (Collaborative 2-F-RA) compared with non-collaborative assessment by novices (2-F-RA). As discussed previously, novices may not be familiar with certain tasks, particularly during the initial stage of work, and thus may have difficulty in relevance-assessment. This work proposes collaborative assessment as a means of addressing this issue. Collaborative assessment aggregates the relevance ratings of evaluators derived from the assessment of experienced users and novices, as addressed in Step 4 of Section 5.3.1. The individual assessments of novices are considered as the non-collaborative assessment to derive the ratings of task relevance.

Table 11 shows the effectiveness of knowledge support for task-retrieval and document-retrieval under collaborative 2-F-RA (by experienced users and novices) and non-Collaborative 2-F-RA (by Novices). The effectiveness is compared according to the average *interpolated precision* at six recall levels and their overall average.

Observation and Implication: The result demonstrates that the effectiveness (precision) of collaborative 2-F-RA method is higher than that of non-collaborative

Table 11. Results of knowledge support
(Non-collaborative 2-F-RA versus collaborative 2-F-RA)

	Task retrieval		Document retrieval	
Recall Level	**Non-C. 2-F-RA**	**Colla. 2-F-RA (E & N)**	**Non-C. 2-F-RA**	**Colla. 2-F-RA (E & N)**
	Precision	Precision	Precision	Precision
[0.0, 0.2)	1.000	1.000	0.800	0.863
[0.2, 0.4)	0.812	0.873	0.226	0.232
[0.4, 0.6)	0.728	0.868	0.209	0.216
[0.6, 0.8)	0.556	0.666	0.174	0.181
[0.8, 1.0)	0.461	0.527	0.149	0.155
[1.0, 1.0]	0.357	0.415	0.137	0.137
6-pt Average	0.652	**0.725**	**0.282**	**0.296**

2-F-RA method. The result reveals that novices can obtain more effective knowledge support through the collaboration from experienced users by the *collaborative relevance-assessment approach*. The collaboration among knowledge workers can mitigate the difficulty of accessing task-relevant knowledge from the knowledge repository. Meanwhile, the retrieval performance of collaborative 2-F-RA is better than that of collaborative 2-F-RA method at each recall level. It indicates the stability of *collaborative relevance-assessment approach*.

5.7 Discussions

The experiments were conducted to evaluate the effectiveness of the proposed *collaborative relevance-assessment approach* (Collaborative 2-F-RA). We conduct three experiments to evaluate the effect of linguistic assessment, two-phase relevance assessment, and collaborative assessment of the proposed approach, respectively. The results demonstrated that the linguistic relevance assessment (F-RA) can provide better knowledge support than the binary assessment (B-RA) and query-based method; the two-phase relevance assessment (2-F-RA) provides better knowledge support than the one-phase relevance assessment (F-RA); and the collaborative relevance assessment (Collaborative 2-F-RA) provides better knowledge support than the non-collaborative relevance assessment (2-F-RA). Novices can obtain more effective knowledge support through the collaboration from experienced users. Although the improvement of adding one more factor is not significant, the improvement of final collaborative 2-F-RA method over Query-based method is significant. The result of the Query-based method (baseline method) is listed in Table 5, and the result of *collaborative relevance assessment* method (Collaborative 2-F-RA) is listed in Table 10. For novices, the Collaborative 2-F-RA achieves 25.86% better than that of Query-based method. The results demonstrate that the proposed *collaborative relevance-assessment approach* can provide effective knowledge support in task-based environments.

Chapter 6 Disseminating and Sharing Task-relevant Knowledge

Effective knowledge management relies on understanding workers' information needs on tasks, for brevity, task-needs. Generally, the worker's task-needs may change over time; therefore, a promising model to track worker's dynamic information needs on task is demanded. An overview of knowledge support ($\mathcal{K}$-Support) model based on profiles to facilitate knowledge dissemination and sharing is first given in this chapter. Accordingly, an adaptive task-based profiling approach is proposed to tackle workers' dynamic information needs on tasks. And then, a fuzzy analytical method is proposed to identify peer-groups with similar task-needs based on workers' profiles. Finally, a task-based knowledge support ($\mathcal{K}$-Support) system is developed to acquire, organize, and disseminate an organization's knowledge resources from the aspect of business task. We also conduct various system evaluations to examine the effectiveness of the proposed model applied in system.

6.1 Overview of $\mathcal{K}$-Support model

Primary concepts of knowledge support model for disseminating and sharing task-relevant knowledge are addressed below.

- Knowledge dissemination and sharing rely on profile modeling to capture workers information needs on the target task. A systematic approach to model the worker's initial task-needs is described in Chapter 5 [85]. In this chapter, two kinds of profiles, feature-based profile and topic-based profile, are proposed. Both profiles are used to represent a worker's current information needs on the target task at hand.

- An adaptive task-based profiling approach is proposed to model workers' dynamic information needs (profiles) on tasks. Task-based knowledge support can then be facilitated to assist knowledge workers to access and disseminate task-relevant knowledge based on the profile, i.e. task profile. A fuzzy linguistic approach is employed to model workers' relevance feedbacks. Moreover, a modified relevance feedback (RF) technique, adopted from the techniques proposed by Rocchio (1971) and Ide (1971), is used to adjust workers' profiles based on relevance feedbacks.

- For complex and knowledge-intensive tasks, collaboration among knowledge workers and experts is often necessary for more effective knowledge dissemination. A fuzzy analytical method is proposed to determine peer-groups with similar task-needs based on the profile, i.e. work profile. The method employs a fuzzy max-min operation to derive the similarity among workers by computing the transitive max-min closure [16][40]. Inherent transitive relationship among workers is inferred to derive a fuzzy similarity matrix of workers. Task-based peer-groups can then be identified by grouping members with equivalence relation determined by α-cuts applied to the fuzzy similarity matrix. The proposed system can provide more effective knowledge support through knowledge sharing among peer-group members. Peer-group members engaged in common tasks or with similar task-needs can collaborate in the proposed task-based portal to accomplish their tasks.
- In task-based environments, codified knowledge and human resources are important knowledge assets for accomplishing organizational tasks. This work presents an architecture and implementation of a knowledge support system ($\mathcal{K}$-Support) in task-based workplaces. The proposed $\mathcal{K}$-Support system provides task-relevant knowledge to a worker based on his/her information needs on the target-task, namely the task being conducted at hand.

Figure 2 in Chapter 3 shows the proposed knowledge support model based on profiles to facilitate task-based knowledge delivery and sharing. Workers' information needs generally change during progress on performing the target task. The *user behavior tracker* in the proposed framework is an on-line module to capture workers' dynamic behaviors, including access behaviors on the task-based domain ontology and relevance feedbacks on knowledge items. The *profile handler* uses an adaptive task-based profiling approach to adjust workers' profiles. The *peer-group analyzer* employs a fuzzy analytical method to identify peer-groups with similar task needs (information needs on the target task) based on work profiles. Section 6.2 and 6.3 will detail the proposed methods.

6.2 Adaptive task-based profiling approach

This section describes the adaptive task-based profiling approach which models workers' dynamic information needs on the target task. The adaptation of profile is based on users' access behaviors or relevance feedback on knowledge items. A modified relevance feedback technique is employed to adjust workers' profiles.

6.2.1 Profile modeling and structuring

We propose an adaptive task-based profiling approach to model workers' dynamic information needs via feedback analysis. Meanwhile, worker's relevance feedback is modeled by a fuzzy linguistic approach, as described in the following.

Perception modeling through fuzzy linguistic approach. The fuzzy linguistic approach is a technique for approximating human perception, and provides easier assess to qualitative problems. Linguistic assessment uses words rather than numbers. For example, the linguistic variable "*Relevance*" is defined to assess the degree of relevance between objects (such as document, task, etc.). Notably, a linguistic variable is characterized by a quintuple (S, $E(S)$, U, G, M) as defined in *Definition I* of Appendix A [91]. The semantic meaning of a linguistic term can be formulated as a fuzzy number, which represents the approximate value of each linguistic term.

Profile structuring. Two kinds of profiles, feature-based task profile and topic-based task profile, are maintained. Both profiles are used to represent a worker's current information needs on the target task at hand.

- **Feature-based task profile:** The feature-based task profile of a task t_r is a feature vector of weighted keywords, denoted as $\vec{t}_r$ = $<w_{kw1}, w_{kw2}, \ldots, w_{kwn}>$. The representation of feature-based task profile is the same as the profile that generated based on the task-relevance assessment result.
- **Topic-based task profile:** The topic-based task profile of a worker u, denoted as $WP_u = \{<topic_j, w_p(topic_j)>\}$, contains a set of topics (fields or tasks in domain ontology) with associated *degree of relevance* to the target task at a specific time period. $w_p(topic_j)$ represents the relevance degree of $topic_j$ to the target task at time p, from the aspect of u. Let $\boldsymbol{FS}$ denote the set of topics in field level and $\boldsymbol{TS}$ denote the set of topics in task level. Note that the category level is not considered since the topics in category are too general to

differentiate workers' task needs. The associated *degree of relevance* indicates a similarity measure between a topic and the target task at a specific time period. The similarity measure is initially obtained from a worker's relevance assessment, and will be updated via a worker's explicit feedback (e.g., relevance rating) or via analyzing a worker's access behaviors and explicit feedback, will be addressed in Section 6.2.2.

A topic-based task profile represents a worker's task-needs expressed as a set of relevant fields or tasks in domain ontology, and can be used to derive a worker's personalized ontology (WPO) on the target task. An ontology threshold value δ can be defined by a worker to generate a worker's personalized ontology on the target task by filtering out irrelevant fields or tasks with relevance degrees below the threshold value. Accordingly, $WPO_u = \{<topic_j, w_p(topic_j)> \mid w_p(topic_j) \geq \delta$ and $topic_j \in \boldsymbol{FS} \cup \boldsymbol{TS} \}$. The result forms a worker u's personalized ontology on the target task.

6.2.2 Profile adaptation based on feedback analysis

Document feedback analysis. A temporal profile, denoted as $\vec{T}emp_{u,p}$, is generated by the *profile handler* to represent a worker u's current information needs on the target task. The temporal file is derived from the feature vectors of those documents accessed by worker u during time period p, as shown in the Eq. 6.1.

$$\vec{T}emp_{u,p} = \frac{1}{\left|D_{u,p}^{\exp}\right|} \sum_{\forall dj \in D_{u,p}^{\exp}} (A_u(d_j) \times \vec{d}_j) + \frac{1}{\left|D_{u,p}^{imp}\right|} \sum_{\forall dj \in D_{u,p}^{imp}} (CV(\tilde{H})^u \times \vec{d}_j) \qquad \textbf{(6.1)}$$

$D_{u,p}^{\exp}$ denotes the set of documents which had been explicitly rated by worker u in conducting the target task during the time period p. $A_u(d_j)$ denotes worker u's crisp rating on the relevance of document d_j to the target task. The crisp rating is derived from the linguistic rating according to the center of area (COA) method described previously. $D_{u,p}^{imp}$ denotes the set of documents which had been browsed and accessed but not been rated by worker u during time period p. A linguistic rating "High" is given by default to represent the relevance degree of unrated documents (implicit feedback). $CV(\tilde{H})^u$ denotes the corresponding crisp value of relevance rating "High" of worker u. Notably, our system will show the description of a document. Thus, we assume that a worker will read the description first to decide if the document is relevant, and then access and browse the document. Accordingly, a linguistic rating "High" is assigned to unrated documents that had been accessed and browsed by the worker.

The similarity between the temporal profile and a topic t_j in the domain ontology can be derived by cosine measure, namely $sim(\vec{T}emp_{u,p}, \vec{t}_j)$. Notably, $sim(\vec{x}, \vec{y}) = \frac{\vec{x} \bullet \vec{y}}{|\vec{x}||\vec{y}|}$.

Profile adaptation. The new feature-based task profile of the target task, denoted as $\vec{S}_{p+1}$ is generated based on Eq. 6.2, which is modified from standard Rocchio (1971) and Ide (1971) algorithms presented in Section 2.2.2. The modification considers the associated relevance degrees of relevant/irrelevant tasks to the target task and the temporal profile derived form the feedback analysis.

$$\vec{S}_{p+1} = \alpha \vec{S}_p + \beta \vec{O} - \gamma \sum_{\forall t_j \in T_n} (1 - w_{p+1}(t_j)) \vec{t}_j$$

$$\vec{O} = \lambda \sum_{\forall t_j \in T_r} w_{p+1}(t_j) \vec{t}_j + (1 - \lambda) \vec{T}emp_{u,p} \tag{6.2}$$

Where $\vec{S}_p$ denotes the feature-based task profile of the target task at time p. Notably, $\vec{\mathbf{s}}_p$ may be an initial task profile derived from the initial assessment result. The $\vec{O}$ denotes the aggregated relevant feature vector of the target task. The aggregation of irrelevant feature vectors is derived from T_n, which is the set of irrelevant tasks. The relevant feature vector $\vec{O}$ is derived based on T_r , the set of relevant tasks, and the temporal file generated from the feedback analysis. Herein, the set of T_n, and T_r are derived from worker's feedback result. $w_{p+1}(t_j)$ denotes the relevance degree (associated weight) of task t_j to the target task. $\vec{T}emp_{u,p}$ denotes the temporal profile derived from the feedback analysis. Meanwhile, α, β, γ are tuning constants. The parameter λ is used to adjust the relative importance of relevant tasks and the temporal profile. Note that there are two alternatives to derive $w_{p+1}(t_j)$.

- Explicit relevance feedback on tasks: $w_{p+1}(t_j)$ denotes the worker u's crisp rating on the relevance of existing tasks t_j to the target task.
- Adjusting relevance degree of tasks by documents feedback.

The topic-based task profile will be adjusted based on the result of feedback analysis. Note that a topic-based task profile records topics (tasks or fields) with associated relevance degree to the target task. The incremental analysis means the system learning the worker's TRTs (set of relevant tasks) and TRITs (set of irrelevant tasks) on target task from the document feedback analysis. The result of document feedback analysis is to generate a temporal profile to represent a worker's task-needs. Thus, the relevance degree between target task and topics (tasks or fields) are obtained by the similarity calculation between temporal profile and topics. Moreover, since the relevance degree will be adjusted across time (increased or decreased), we named it as an incremental analysis procedure. The method to adjust associated relevance degree of each topic is addressed as follows.

The system will increase or decrease the relevance degree (associated weight) of a task t_j (a topic in the task level of domain ontology) gradually, where $w_{p+1}(t_j)= w_p(t_j)\pm\Delta w$. The adjustment Δw of a task t_j is derived based on the proportion of feedbacks and the similarity between the temporal profile and t_j. If $sim(\vec{T}emp_{u,p},\vec{t}_j)$ is above a relevance-adjustment threshold θ, the system will increase the associated weight of task t_j. Meanwhile, if $sim(\vec{T}emp_{u,p},\vec{t}_j)$ is below θ, the system will decrease the associated weight of task t_j. The adjustment equation is given below.

$$\Delta w(t_j)=\frac{N_{u,p}^d}{N_u^d+N_{u,p}^d}\times\left|sim(\vec{T}emp_{u,p},\vec{t}_j)-\theta\right| \quad \textbf{(6.3)}$$

where N_u^d denotes the number of documents accessed and browsed by worker u in conducting the target task prior to time p, while $N_{u,p}^d$ denotes the number of documents accessed and browsed by worker u in conducting the target task during time p. Notably, $w_{p+1}(t_j)=1$, if $w_p(t_j)+\Delta w>1$; $w_{p+1}(t_j)=0$, if $w_p(t_j)-\Delta w<0$. Moreover, a field contains a set of tasks. Thus, the value of $w_p(field_i)$ is set to the maximum value of $w_p(t_j)$ for any *task* t_j belongs to $field_i$. Namely, the weight of $field_i$ will be adjusted at time $p+1$, where $w_{p+1}(field_i) = max_{tj\in fieldi}(w_{p+1}(t_j))$. Thus, it is a incremental analysis procedure to calculate the relevance degree of topics to target task over time. Meanwhile, the adjustment may change the information structure of a worker's personalized ontology. The personalized ontology of worker u is adjusted by removing an irrelevant topic t_j , if $w_{p+1}(t_j)$ is below the ontology threshold δ, and adding a relevant topic t_j , if t_j did not exist at time p and $w_{p+1}(t_j)\geq\delta$. Accordingly, this approach is different from the method of explicit feedback. The degree of relevance of a topic, $w_p(topic_j)$, in the work profile is inferred from incremental analysis instead of explicit feedback on tasks, i.e., relevance rating.

Furthermore, the feature-based task profile of the target task can be adapted based on the adjustment of topic-based task profile. The system generates a set of top-*k* relevant tasks (denoted as TRTs) and a set of top-*k* irrelevant tasks (denoted as TIRTs) based on the topic-based task profile. Relevant tasks are those tasks with associated relevance degree $w_{p+1}(t_j)$ higher than the relevance-adjustment threshold θ, whereas irrelevant tasks are those tasks with $w_{p+1}(t_j)$ lower than θ. Only the feature terms and the associated relevance degrees of topics in the task level are used to adjust the task profile of the target task. A field is a generic view of similar tasks; hence, the feature terms of fields are not as representative as the feature terms of tasks for the target task. The new feature-based task profile of the target task, denoted as $\vec{S}_{p+1}$ is also generated based on Eq. 6.2.

The advantage of this method, i.e., adjusting relevance degree of tasks by document feedback, is that it dose not require workers to conduct tedious relevance feedback explicitly. However, this method need to continuously track and record workers' access behaviors. Moreover, the time factor is also important to analyze the worker's access behaviors. More recent access behaviors should give higher weight than earlier access behaviors in adjusting the relevance degree of tasks. Thus, further study is needed to investigate and evaluate this method.

6.2.3 K-Delivery: Delivering codified knowledge proactively

A worker's feature-based task profile and topic-based task profile can properly reflect a work's task-needs on the target task. The profiles can be used to further enhance the knowledge retrieval capability in the proposed system. Moreover, the adjustment of work profile across time will lead the system to refine the task profile based on the proposed profile adaptation approach. Accordingly, $\vec{S}_{p+1}$ is used to retrieve relevant codified knowledge in the repository. Relevant task and document sets will be retrieved to (e.g. cosine measure). Figure 8 is the interface of knowledge delivery in which the system delivers task-relevant knowledge proactively based on the task profiles.

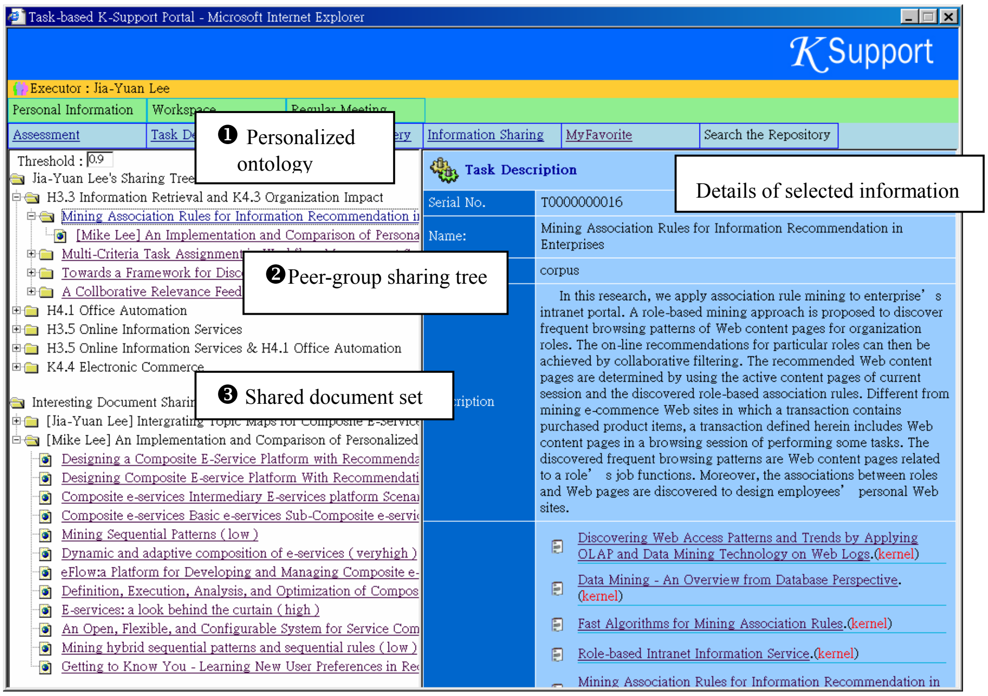


Fig. 8. Interface of knowledge sharing (α=0.9)

6.3 Peer-group analytical model

This section discusses the proposed similarity analytical model to identify peer-groups with similar task needs based on topic-based task profiles. Notably, the topic-based task profile records a worker's task-needs on the target task, which are represented as a set of topics (fields or tasks in domain ontology) with associated degree of relevance to the target task. The proposed method mainly contains two phases. In phase 1, a user-user similarity matrix is constructed to record workers' similarity relationships on task-needs. In phase 2, a fuzzy inference procedure is employed to infer the implicit and transitive relationships among workers. The α-cuts approach is then applied to generate a proper set of task-based peer-groups.

6.3.1 Establishing a user-user similarity matrix

The similarity measure between workers E_x and E_y can be derived using the associated relevance degrees $w_p(topic_j)$ of topics recorded in the topic-based task profiles of E_x and E_y. The task-level/field-level relevance degrees can be used to derive the task-level/field-level similarity between workers. As described in Section 6.2.2, the relevance degree in field level is derived from the task-level, namely, the value of $w_p(field_i)$ is set to the maximum value of $w_p(t_j)$ for any *task* t_j belongs to

$field_i$. Thus, the task-level similarity is stricter than the field-level similarity. Very few similar users can be identified based on the task-level similarity if there are very few tasks relevant to the target task. More number of similar users can be identified based on the field-level similarity. However, our experimental analysis shows that the field-level similarity derived from the maximum value of task-level relevance degree is too vague to measure the similarity between workers. Accordingly, we employ a compromised approach to compute field-level similarity based on the aggregation of task-level relevance degrees. An aggregated field-level relevance degree of a $field_i$ is derived from the aggregation (summation) of $w_p(t_j)$ for all *task* t_j belongs to $field_i$. Then, the field-level similarity is measured according to the aggregated field-level relevance degrees, as described in the following steps.

Step 1: Constructing a task-level user feedback matrix

An n-by-k user-feedback matrix ***I*** *(task-level)* is constructed to represent each worker's task-level relevance degrees recorded in each worker's topic-based task profile, where n denotes the number of workers, and k denotes the number of task items. The task-level relevance degrees represent workers' perspective on the relevance of tasks to the target task.

Step 2: Deriving a field-level feedback matrix

An n-by-l user-feedback matrix I (field-level) is derived via a matrix operation employed between the transpose of an l-by-k field-to-task binary relationship matrix F (described in Section 4.1) and task-level user-feedback matrix I (task-level), as shown in Eq. 6.4.

$$I(field-level)_{n-by-l} = I(task-level)_{n-by-k} \times F^T_{k-by-l} \qquad \textbf{(6.4)}$$

l denotes the dimension of *Field*, k denotes the dimension of *Task*, and n denotes the number of workers.

Step 3: Determining the similarity relationship matrix

The cosine measure (Eq. 6.5) is employed to calculate the similarity among workers based on the n-by-l user-feedback matrix I (field-level).

$$\varsigma(E_i, E_j) = \frac{\vec{A}_{E_i} \bullet \vec{A}_{E_j}}{\left|\vec{A}_{E_i}\right|\left|\vec{A}_{E_j}\right|} \qquad \textbf{(6.5)}$$

$\vec{A}_{E_j}$ and $\vec{A}_{E_j}$ are workers' E_i and E_j's feedback values in field-level derived from the user-feedback matrix ***I*** (field-level).

Finally, a reflective and symmetric matrix is derived, denoted as an *n*-by-*n* fuzzy similarity relationship matrix ***S***, which represents the similarity-relationship on workers' task-needs.

$$\boldsymbol{S} = \begin{bmatrix} 1.00 & 0.90 & 0.56 & 0.00 \\ 0.90 & 1.00 & 0.64 & 0.00 \\ 0.56 & 0.64 & 1.00 & 0.56 \\ 0.00 & 0.00 & 0.56 & 1.00 \end{bmatrix}$$

6.3.2 Identifying task-based peer-groups

A fuzzy inference procedure is employed to infer the implicit and transitive relationships among workers. The α-cuts approach is then applied to generate a proper set of task-based peer-groups. This section demonstrates how the task-based peer-groups can be automatically identified. Notably, the fuzzy inference procedure is used to derive the inherent transitive relationships among workers. Accordingly, the system can identify similar workers with implicit and inherent transitive relationships, even if very few explicit similarity measures are found in the similarity relationship matrix ***S***.

Step 1: Inferring user relationship by fuzzy inference

The *n*-by-*n* similarity relationship matrix ***S*** represents the similarity relation among ***U***, a set of workers. The relation of workers is represented in terms of membership function $\tilde{\varsigma}(E_i, E_j) \in [0,1]$. The method of transitive max-min closure [16][40][42] is adopted to derive a reflective, symmetric, and transitive matrix, which is a equivalence matrix. The definition of a transitive max-min closure $\boldsymbol{S_T}$ of the similarity matrix ***S*** is defined in *Definition III* of Appendix B*,* which is adopted from [40].

Assume that the initial similarity relationships of workers are shown in the left part of Figure 8. The right part shows the inferred transitive relationships among workers in matrix $\boldsymbol{S_T}$ after transitive max-min operations. The dashed line indicates the new inferred relationships after transitive max-min operations. For example, the relationship between E_i and E_l is 0.56.

$$
S_T = \begin{bmatrix} 1.00 & 0.90 & \mathbf{0.64} & \mathbf{0.56} \\ 0.90 & 1.00 & 0.64 & \mathbf{0.56} \\ \mathbf{0.64} & 0.64 & 1.00 & 0.56 \\ \mathbf{0.56} & \mathbf{0.56} & 0.56 & 1.00 \end{bmatrix}
$$

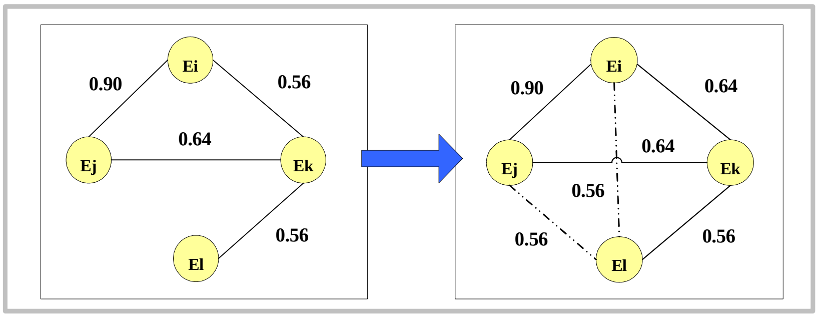


Fig. 9. Inferring similarity relationships based on workers' task-needs

Step2: Identifying task-based peer-group by α-cuts.

The α-cuts can be applied to the equivalence matrix $\boldsymbol{S_T}$ for any α degree to group workers in $\boldsymbol{U}$, where $\alpha \in (0,1)$. Workers grouped together have equivalence relation. Several α degrees can be gradually refined to partition workers to form the subsets with equivalence relations. Different subsets of equivalence relations are derived by setting different α degrees in the matrix $\boldsymbol{S_T}$ to partition set $\boldsymbol{U}$. For example, two subsets of equivalence relations are derived by setting α=0.64 in the matrix $\boldsymbol{S_T}$ to partition set $\boldsymbol{U}$.

$$
S_T^{\alpha=0.64} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \alpha=0.64
$$

6.3.3 K-Sharing: Knowledge support from peer-group

The *K*-Support system uses topic-based task profiles to identify task-based peer-groups. Two kinds of task-based peer-groups are located. The first is formal task-related members who join the same projects. The other is informal peer-groups with similar task-needs identified by the system. The system not only provides a knowledge support platform for gathering and exchanging task-relevant knowledge among workers, but also presents the peer-group member's personalized ontology on the target task for knowledge sharing.

6.4 Task-based $\mathcal{K}$-Support portal

The task-based $\mathcal{K}$-Support portal is a Web-based application, allowing workers to retrieve, organize and share task-relevant knowledge.

Delivering codified knowledge proactively. Weighted discriminating terms are kept in the task profile for retrieving task-relevant knowledge. The system can recommend task-relevant information based on the worker's profiles. A tree-like structure is employed to organize the task-relevant information. Once the worker selects a document or a task topic to read, the detailed information will be displayed. The worker can download the document or store it to the *MyFavorite* folder. The *user behavior tracker* in the profile modeling server will track the worker's feedback or access behavior to adapt the feature-based task profile and topic-based task profile.

Knowledge-sharing from peer-group. The system expands the ontology of a worker with the peer-group member's personalized ontology for knowledge sharing. The system facilitates knowledge sharing by displaying the shared information such as tasks and documents from task-based peer-groups. The left frame of Figure 8 shows the sharing tree of "Jia-Yuan Lee". A sharing tree is a tree-like structure, which represents the personalized ontology of a worker. Meanwhile, the shared information from task-based peer-groups is also presented in the sharing tree. All information is calculated according to the feedback results. A threshold, α-cut level, which is shown in the top of the left frame, can be adjusted by the workers to find more peer-group members by decreasing the α value.

6.5 Experimental setup

Various experiments are conducted to evaluate the effectiveness of the proposed $\mathcal{K}$-Support system. Evaluation is conducted with respect to the information needs of users participating in knowledge-intensive tasks such as conducting thesis work or research projects.

6.5.1 Overview of experiments

Experimental objective and design

The $\mathcal{K}$-Support system consists mainly of two applications, *K-Delivery* and *K-Sharing*. The *K-Delivery* application delivers task relevant knowledge to workers based on the adaptation of feature-based task profiles. Notably, the task profiles are adapted according to worker's dynamic information needs, namely access behaviors or explicit feedback, as described in Section 6.2. The *K-Sharing* application provides peer-group members' task relevant knowledge for knowledge sharing, as described in Section 6.3. Accordingly, we evaluate the effectiveness of *K-Delivery* application by comparing the *K-Delivery* based on initial feature-based task profiles (without adaptation) with the *K-Delivery* based on adapted feature-based task profiles. Moreover, we evaluate the quality and novelty of shared knowledge items provided by the *K-Sharing* application. That is, two evaluation metrics were considered to examine the effectiveness of the system-- the *novelty* and the *quality* of the knowledge items provided from the system.

The *K*-Support system contains three phases, including *phase I*: *K-Delivery* based on initial feature-based task profiles, *phase II*: *K-Delivery* based on adapted feature-based task profiles, and *phase III*: *K-Sharing* from task-based peer-groups. The *K*-Support system performs phase I, then phase II, and finally phase III. We evaluate the effectiveness of profile adaptation based on the experimental result of phase II. The evaluation will demonstrate whether the proposed adaptive task-based profiling method can model worker's dynamic information needs properly. We also evaluate the effectiveness of knowledge sharing based on the experimental result of phase III. The evaluation will demonstrate whether the proposed peer-group analytical model can effectively identify task-based peer-groups based on topic-based task profiles.

6.5.2 Data, participants and evaluation metrics

Data and Participants

Experiments were conducted using data collected from a research institute. Forty-eight research tasks were collected, with 32 existing-tasks and 16 executing-tasks (target tasks that workers conduct at hand). Six executing-tasks were chosen as the testing set for evaluations. Over 500 task-related documents were collected. Tasks are classified into five categories and then grouped into thirteen fields. The smallest meaningful components of document information elements, such as title, abstract, journal and author, were extracted from documents. Each document contained an average of ninety distinct terms after information extraction, and document pre-processing (e.g. case folding, stemming, and stop word removal).

Twelve users were selected to participate in the evaluation. Two kinds of user group were selected to conduct the experiments. One group consisted of experienced users who were familiar with the executing task and the other group consisted of novices who were unfamiliar with the executing task.

Performance evaluation metrics

In general, evaluating the retrieval performance by considering all retrieved items is difficult, since users may not give feedback values (relevance ratings) on all retrieved items. User-oriented metrics derived from users' perceptions on retrieved items were usually used to evaluate the retrieval performance [7]. Two user-oriented metrics, novelty and quality, were adopted to evaluate the effectiveness of the proposed system from users' perceptions on knowledge items, namely users' feedback values (relevance ratings) on knowledge items. The retrieved knowledge items which have been rated by users were used to derive the evaluation metrics. Moreover, the evaluation considered three phases of the system.

The *novelty* metric measures the ratio of relevant knowledge items retrieved that are unknown to the user (worker) E_i, as defined in Eq. 6.6. The relevant knowledge items are those items retrieved with feedback value above "Normal" from worker's perception.

$$Novelty = \frac{|R_u|}{|R_u| + |R_k|} \tag{6.6}$$

where $|R_k|$ denotes the number of relevant knowledge items retrieved (in current and previous phases) which are known to worker E_i, whereas $|R_u|$ denotes the number of relevant knowledge items retrieved (in current phase) which are unknown to worker E_i. Notably, an item is known (unknown) to worker E_i, if that item had (not) been rated by E_i in previous phases. The *novelty* metric is used to measure the effectiveness of the system in discovering new (previously unknown) knowledge items that suit user needs.

The *quality* metric measures the fraction of aggregated ratings of retrieved knowledge items to the aggregated maximum ratings of retrieved knowledge items, as defined in Eq. 6.7.

$$Quality = \frac{\sum_{j \in R} a_j^{E_i}}{CV(\tilde{P})^{E_i} \times |R|} \quad \textbf{(6.7)}$$

$CV(\widetilde{P})^{Ei}$ denotes the corresponding crisp value of maximum relevance rating "Perfect " given by worker E_i. R denotes the set of knowledge items retrieved and rated by worker E_i in current phase. a_j^{Ei} is the crisp feedback value on retrieved knowledge item j given by worker E_i. The *quality* metric is used to measure the worker's satisfaction degree on the retrieved knowledge items (e.g. tasks and documents).

6.6 Experimental results and implications

6.6.1 Novelty of knowledge support

Table 12 shows the novelty of *K-Delivery* and *K-Sharing*, respectively. Notably, the novelty for Initial *K-Delivery* in phase-I is not filled out since the novelty is 1.000, i.e., all retrieved knowledge items are unknown to users in *initial K-Delivery*.

Observations: The result shows that *K-Delivery* in phase-II can discover new (unknown) and relevant (feedback value above "Normal") items based on adapted task profiles. Thus, the adaptation of task profiles to model workers' dynamic task-needs is important to provide necessary knowledge support. Moreover, Table 12 shows that the novelty of *K-Sharing* in phase-III is higher than that of *K-Delivery* in phase-II. The result reveals that *K-Sharing* in phase-III can help workers find more new and relevant knowledge items from peer-group members.

Implications: The novelty of task-items under *K-Delivery* in phase-II is below 0.5. The result implies that the relevant task set is stable from phase-I to phase-II. Furthermore, for experienced workers, the novelty of task-items under *K-Delivery* in phase-II is lower than that for novices. The result implies the task profiles of experienced workers are more stable than those of novices. Novices are usually uncertain about their information needs in the beginning, and thus often adjust their

Table 12. Users' perceptions of information novelty (Average novelty)

Phases of *K*-Support	Conditions	Experienced		Novices	
		Task	Document	Task	Document
Phase I	Initial K-Delivery	--	--	--	--
Phase II	Adapted K-Delivery	0.283	0.540	0.373	0.520
Phase III	K-Sharing	0.612	0.570	0.650	0.613

information needs during task performance. The *adapted K-Delivery* can find more proper relevant tasks for novices based on the adaptation of task profiles.

6.6.2 Quality of knowledge support

Table 13 shows the quality of *K-Delivery* and *K-Sharing*, respectively. All three phases can provide workers knowledge items that suit their needs.

Observations: In general, the quality (satisfaction degree) of *K-Delivery* in phase-II is higher than that of the other two phases. *K-Delivery* in phases-II shows good adaptation capability to satisfy workers' needs based on adapted task profiles. Interestingly, the quality for novices is better than experienced workers, especially in phase-II and phase-III.

Implications: The result indicates that the *K*-Support system can provide workers appropriate and needed knowledge items based on the adaptive task-based profiling approach. We observed that experienced workers are more knowledgeable on the executing-tasks, thus are more certain on the relevance of knowledge items; most novices are not knowledgeable on the executing tasks, thus are uncertain on the relevance of knowledge items, and tend to give relevant ratings.

The overall experimental results conclude that the adaptive task-based profiling method and the fuzzy peer-group analytical model are effective to stimulate knowledge retrieval and knowledge sharing.

Table 13. Users' perceptions of information quality

Phases of *K*-Support	Conditions	Experienced		Novices	
		Task	Document	Task	Document
Phase I	Initialized K-Delivery	0.703	0.639	0.702	0.621
Phase II	Adapted K-Delivery	0.657	0.773	0.784	0.774
Phase III	K-Sharing	0.569	0.725	0.689	0.767

6.7 Discussions

A *K*-Support portal is built upon the system to facilitate task-based knowledge retrieval and sharing among task-based peer-groups. The knowledge support is realized by the proposed profile modeling approach. Therefore, the problem of accessing needed knowledge items from vast amounts of codified knowledge can be alleviated. In addition, this system identifies task-based peer-groups based on the proposed fuzzy analytical method. Knowledge sharing is achieved by enabling workers to share their task-relevant knowledge among peer-groups.

Several issues need further investigations. First, alternative methods to analyze worker's task-needs on topics by incremental analysis need to be evaluated. Second, the information needs of knowledge workers are associated with their roles in undertaken tasks; however, this work does not consider the role/job perspective [5] to acquire and disseminate task-relevant knowledge. Moreover, a more elaborate profiling approach, which considers the characteristics of business tasks and the dynamic long-term task-needs, is also demanded. Future studies could extend the proposed profiling approach to acquire and reuse corporate memory effectively.

Chapter 7 Task-Stage Knowledge Support

According to our empirical investigation, a worker has different information needs during the long-term task performance. Although we have proposed a profiling method in previous work [47] to learn user's dynamic needs; however, workers still suffer the problem of finding the pertinent information, which tightly reflect their current task-needs. We analyze the problem from twofold: (1) The characteristic of knowledge retrieval activity in a working environment is that the worker's information needs are associated with the executing task at hand; (2) Meanwhile, a knowledge-intensive task consists of levels of progressively smaller subtasks to achieve the main task goal. This chapter extends *adaptive task-based profiling approach* to provide task-relevant knowledge based on worker's task-needs and task-stage. A task-stage knowledge support model is proposed to provide effective knowledge support by identifying a worker's information needs at various task-stage. A *correlation analysis technique* is proposed to determine a worker's task-stage (e.g., pre-focus, focus formulation, and post-focus task stages). Meanwhile, an *ontology-based topic discovery technique* is proposed to examine the variety of a worker's task-needs for specific topics within the domain ontology (DO). Empirical experiments are conducted to evaluate the effectiveness of the proposed task-stage knowledge support model.

7.1 Task-needs evolution pattern modeling

7.1.1 Task-stage Knowledge Support Module

There are three phases in providing pertinent task stage knowledge, namely, data pre-processing, task-needs discovery, and adaptive task-stage knowledge router. Note that the task-needs evolution discovery phase is the kernel of the system for analyzing the worker's task stage and task-need topics of stages.

Two types of valuable information: content data and usage data are acquired during *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 can be captured and recorded into the profiles, including the access behavior on the task-based domain ontology and relevance feedbacks on knowledge items. The profile handler uses an adaptive task-based profiling approach to adjust workers' profiles based on the workers' dynamic behavior. The operation details of the adaptive task-based profiling approach can be found in our previous work [47]. The *task-need evolution discovery phase* is the kernel of the system for analyzing the worker's task-related behavior during task performance, as shown in Figure 10. The task-stage identifier and the task-need analyzer are within the task-needs evolution module for identifying the worker's task stage and analyzing task-need topic of each stage based on the variety of profiles. Herein, worker's task-needs are modeled as the topic nodes in DO at different abstraction level which are relevant to the target task.

- **Task-Stage Identifier:** The task-stage identifier is responsible for analyzing and determining worker's task stage based on the changes of the task profile over time.

- **Task-Need Analyzer:** The task-need analyzer is responsible for tracking the worker's access behavior over a period of time. The access behavior is analyzed based on the domain ontology (DO) to discover worker's task-needs on specific topics. Herein, the DO is a multi-level structure and each node in the DO represents a research topic in our application domain, as shown in Figure 3 of Section 4.2.

The IF strategy can be adopted to provide stage-relevant knowledge based on the analyzing results of task-needs evolution phase, namely, stages and stage topics. Therefore, the adaptive task-oriented knowledge router could provide workers

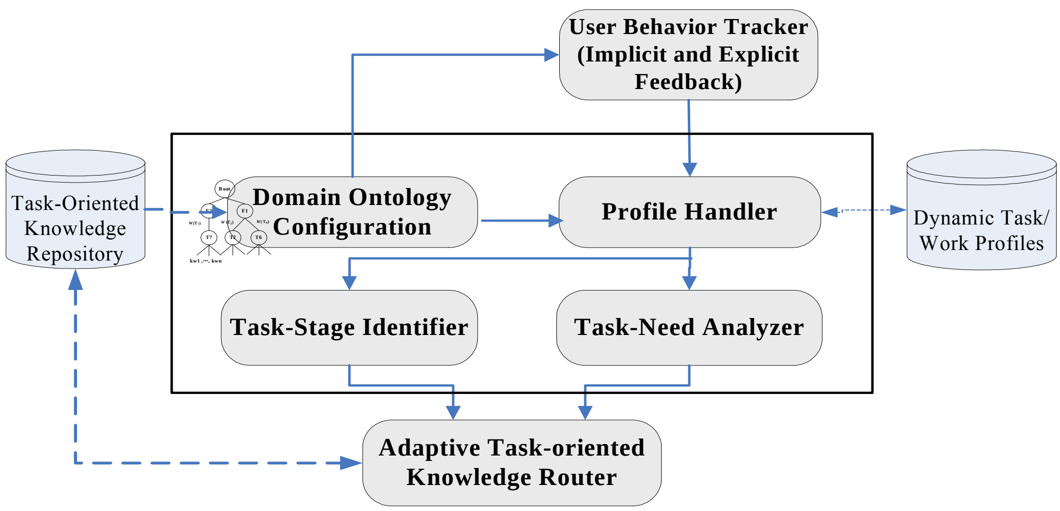


Fig. 10. Task need evolution module

needed and pertinent task-relevant codified knowledge that considers the workers' current task-stage and the needed topics at each stage.

7.1.2 Task-needs evolution pattern modeling

The user behavior tracker tracks and records the worker's access behavior over a period of time when the worker logs into the system. A user-task *session* and *transaction* are defined in this work to analyze workers' implicit and explicit feedback behaviors on codified knowledge items periodically. A *session* is defined as a sequence of user feedback behavior (e.g., reading, downloading or rating an information item) during a single visit to the system. Furthermore, the *task transaction* records the worker's access to the knowledge repository across sessions. In other words, a worker's task *transaction* comprises *n* sessions, where $n \geq 0$. The time interval of a transaction is based on the characteristics of our research application domain, in which the user behavior tracker is activated to generate or update profiles once a worker has uploaded behavior for a specific task.

The worker's session or transaction temporal profile is generated based on the tracking result over a time period of session or transaction. Furthermore, the system analyzes the relevance degree between session or transaction temporal profile and topics in domain ontology (DO). Consequently, a worker's task-needs pattern is expressed in terms of set of topics that are the field-level or 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.

Example 1: In the given example, three sessions are identified in the third transaction of executor "*PoTsun*" who is the executor of "*ITIL-based: Context-aware Knowledge Recommendation*" task. As we have mentioned previously, the time interval of a transaction in our research application domain is that the worker uploads the task-relevant information to the system. Accordingly, the third transaction means the third upload information behavior by the executor "*PoTsun*" for a specific task. Meanwhile, after conducting data preprocessing in phase one, a set of the worker's access behavior patterns across sessions, $Trans_i=\{s_1, s_2, \ldots, s_m\}$, and a set of accessed knowledge items, $O=\{I_1, I_2, \ldots, I_n\}$, are identified.

$$AI(Trans_3^{S_1}) :< I_{368} >$$

$$AI(Trans_3^{S_2}) :< I_{376,} I_{458,} I_{376,} I_{375,} I_{460} >$$

$$AI(Trans_3^{S_3}) :< I_{376}, I_{461}, I_{462}, I_{375}, I_{368}, I_{376} >$$

$AI(Trans_i^{S_j})$ represents a sequence of knowledge items accessed in session *j* of transaction *i*. Our user's access behavior includes explicit feedback behavior, such as rating and uploading, and implicit feedback behavior, such as browsing and downloading. For example, the implicit and explicit behavior for knowledge item $\boldsymbol{I_{376}}$ occurs at different times of the same session.

Task-needs patterns: The temporal profile is derived from the feature vectors of those documents accessed by a worker over a time period. In this work, $\overrightarrow{Trans_i^{S_j}}$ denotes the temporal profile (feature vector of weighted terms) derived from the documents accessed in session *j* of transaction *i*. And then we take further analysis by calculating the similarity (e.g., the cosine measure) between the temporal profile, $\overrightarrow{Trans_i^{S_j}}$ and the profile of a topic, $\overrightarrow{topic_j}$. Notably, $\overrightarrow{topic_j}$ represents the associated profile (feature vector of weighted terms) of $topic_j$. Note that the topic *j* represents a research topic, which is the node in the proposed multi-level domain ontology (DO). Accordingly, a worker's task-needs pattern can be expressed as a set of topics and associated relevance degree. The task-need pattern of a session *j* in transaction *i* is denoted by $Patt_{Trans_i}^{s_j}$. $Patt_{Trans_i}^{s_j}$ is expressed as a set of topics with the associated relevance degree ($topic_j$, rd_j).

Top task-relevant topics: Furthermore, we set a threshold δ or top-N to select the top task-relevant topics (denoted as TRTs) from the task-needs pattern, $Patt_{Trans_i}^{s_j}$. Let *TRTWs* denote the set of top relevant topics with the associated weight derived by the similarity calculation. *TRTWs* is expressed as a set of (*task_id*, *relevance degree*) pairs. Accordingly, the *TRTWs* with associated degree of relevance are recorded in the worker's topic-based task profile to model his/her task-needs to the target task over a time period (regarding a transaction/session).

Example2: A set of TRTs with the associated weight is derived by the similarity calculation and is expressed by (*task_id*, *relevance degree*) pair. The top-4 task-relevant topics of each session within transaction 3 are listed below.

$$TRTWs^t(Trans_3^{S_1}) = \{(t_{07}, 0.140), (t_{50}, 0.123), (t_{12}, 0.100), (t_{31}, 0.097)\}$$
$$TRTWs^t(Trans_3^{S_2}) = \{(t_{48}, 0.115), (t_{18}, 0.096), (t_{47}, 0.096), (t_{23}, 0.092)\}$$
$$TRTWs^t(Trans_3^{S_3}) = \{(t_{50}, 0.128), (t_{07}, 0.119), (t_{19}, 0.116), (t_{20}, 0.105)\}$$

$TRTWs^t(Trans_i^{s_j})$ denotes the top task-relevant topics in session *j* of transaction *i*. The superscript *t* in $TRTWs^t(Trans_i^{s_j})$ denotes the task-level topics of DO. We further express the top task-relevant topics to the field-level topics of DO by (*field_id*, *relevance degree*) pairs. For example, task 7 and task 50 both belong to field 8, as shown in Figure 2. Therefore, the top task-relevant topics can be aggregated to the field-levels listed below.

$$TRTWs^f(Trans_3^{S_1}) = \{(f_{08}*, 0.140), (f_{08}*, 0.123), (f_{06}*, 0.100), (f_{01}*, 0.097)\}$$
$$TRTWs^f(Trans_3^{S_2}) = \{(f_{08}*, 0.115), (f_{10}*, 0.096), (f_{01}*, 0.096), (f_{08}*, 0.092)\}$$
$$TRTWs^f(Trans_3^{S_3}) = \{(f_{08}*, 0.128), (f_{08}*, 0.119), (f_{01}*, 0.116), (f_{08}*, 0.105)\}$$

From the example, it is easy to see that the worker's task-needs focus increasingly on the topic of f8, i.e., "*Office Automation*". The example is a simple case to explain the basic process of usage pattern modeling. Based on the proposed idea, the system can track and identify the worker's task-needs on specific topics from different abstraction level of multi-level structure of DO.

7.2 Changes of task-stage

In this section, a *correlation analysis method* is proposed to identify the changes of worker's task stage is proposed in this section. The objective of task stage identification is to identify the worker's task stage and then deliver task-relevant knowledge according to the worker's task-needs based on different stages. In general, three task stages based on the previous pilot studies: task pre-focus, task focus formulation, and task post-focus stages, are identified to differentiate the worker's three types of information needs during task performance [76][77].

7.2.1 Stage identification process

The on-line task-stage identifier analyzes and determines the 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 task temporal profile of the worker's consecutive transactions. We now discuss the three steps that are executed to analyze and determine the worker's task stage.

Step 1. Task-need pattern calculation: As we have addressed in Section 7.1.2, a worker's task-need pattern can be expressed as a set of topics and associated relevance degree. The task-need pattern of a session j in transaction i, $Patt_{Trans_i}^{s_j}$, is expressed as a set of topics with the associated relevance degree ($topic_j$, rd_j). The relevance degree of a topic j, rd_j, is derived by the cosine measure, $sim(\overrightarrow{Trans_i^{S_j}}, \overrightarrow{topic_j})$. The cosine measure of feature vectors is used as similarity measure.

- **Similarity measure:** The cosine formula is a widely used similarity measure to assess the degree of similarity between two items x and y by computing the cosine of the angle between their corresponding feature vectors $\vec{x}$ and $\vec{y}$, which is given by Eq. 2.2. The degree of similarity is higher if the cosine similarity is close to 1.0.

$$sim(x,y) = cosine(\vec{x},\vec{y}) = \frac{\vec{x} \bullet \vec{y}}{|\vec{x}||\vec{y}|}$$

$\overrightarrow{Trans_i^{S_j}}$ denotes the temporal profile (feature vector of weighted terms) derived from the documents accessed in session j of transaction i. Appendix A shows an example of $Patt_{Trans_i}^{s_j}$ and $Patt_{Trans_i}$. $Patt_{Trans_i}$ is defined and generated similarly by considering the whole transaction.

Step 2. Correlation calculation: Once the similarity pattern has been derived, the correlation of the worker's task-need patterns across transactions can be calculated by *Pearson's correlation coefficient*. A reasonable assumption is that the worker's task-needs will not dramatically change during consecutive sessions of the same transaction. Accordingly, we calculate the correlation between the previous transaction, $Trans_{i-1}$, and the start session of current transaction, $Trans_i^{s_1}$, as shown in the equation below. Note that the time interval of a transaction is based on the characteristics of our research domain.

$$corr_u(A,B) = \frac{\sum_{j \in task\ set} (rd_j^A - \overline{rd^A})(rd_j^B - \overline{rd^B})}{\sqrt{\sum_{j \in task\ set} (rd_j^A - \overline{rd^A})^2 \sum_{j \in task\ set} (rd_j^B - \overline{rd^B})^2}} \quad \textbf{(7.1)}$$

Let A represent $Patt_{Trans_{i-1}}$ and B represent $Patt_{Trans_i}^{s_j}$. rd_j^A and rd_j^B are the relevance degree of topic j in the $Patt_{Trans_{i-1}}$ and $Patt_{Trans_i}^{s_j}$, respectively.

The changes of worker's task stage are based on the correlation results of similarity pattern calculation.

Step 3. Task stage determination: The rationale behind the proposed *correlation analysis method* is to identify the worker's task stage based on the changes of task-needs for topics, i.e., the task-need pattern in our work. In other words, some task-relevant topics within the DO may have a high degree of relevance to the temporal profile of the previous transaction, whereas some task-relevant topics 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. We took around one to two years to observe the worker's access behavior based on the correlation analysis method. And three correlation ranges are set based on the result of our sample analysis, which are "low", "moderate", and "high" correlation.

***Low* correlation:** If a worker is in the early stage 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, 0.5), which indicates that he/she is towards or 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 [0.5, 1.0], which indicates the worker dedicates his/her task-needs for specific topics. In other words, the worker accessed and read similar documents belonging to the related topics. Hence, the worker is within task post-focus stage.

Table 14. Task stage identification rule

Input: $corr_u(Patt_{Trans_{i-1}}, Patt^{S_1}_{Trans_i})$: 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** $corr_u(Patt_{Trans_{i-1}}, Patt^{S_1}_{Trans_i})$ **is "**low"

  $Trans_i$ .stage = task pre-focus stage

  Else *$Trans_i$ .stage* = task formulation stage;

"formulation stage": **If** $corr_u(Patt_{Trans_{i-1}}, Patt^{S_1}_{Trans_i})$ **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*)

In this work, the time point to decide the worker's task stage is the beginning of the current transaction, $Trans_i$, which is the worker's current session. Assume we know the worker's task stage for the previous transaction, $Trans_{i-1}$, we can then infer the worker's current task stage from the correlation value. The task stage determination rule is given in the Table 14. In the following, we will explain the worker's evolution pattern based on this rule.

7.2.2 Sample analysis

Example 3: In our domain, there are 36 task-level topics (historical task corpora). Continuing with Examples 1 and 2 in Section 7.1, we analyze the four transactions of the access pattern of the executor "PoTsun". $Trans_2$, $Trans_3$, $Trans_4$ and $Trans_5$ are sampled to explain how the worker's task stage can be determined, as shown in Figure 11.

(1) $Trans_2$ and $Trans_3$: The correlation between $Trans_2$ and the start session of $Trans_3$ is -0.024 (low), which indicates there is a very different pattern between transactions. Because the worker's task stage of $Trans_2$ is the pre-focus stage, we can infer that the worker's task stage of $Trans_3$ is also the pre-focus stage.

(2) $Trans_3$ and $Trans_4$: The correlation between $Trans_3$ and the start 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 of $Trans_4$ is the task formulation stage.

(3) $Trans_4$ and $Trans_5$: The correlation between $Trans_4$ and the start session of $Trans_5$ is 0.705 (high), which indicates that there are very similar patterns between transactions. Because the worker's previous task stage is in 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.

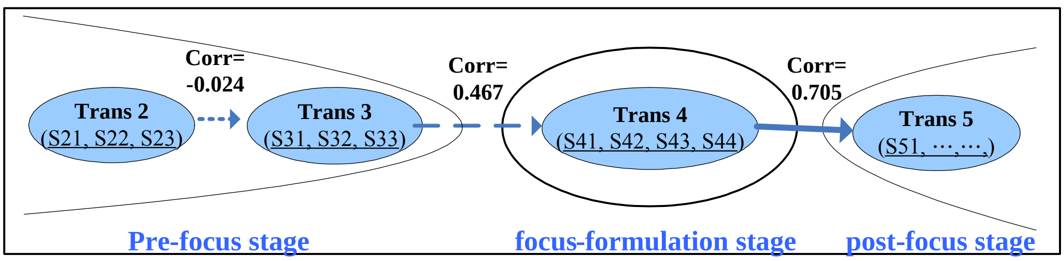


Fig. 11. Changes of task stages

7.3 $\mathcal{K}$-Support based on task-stage and task-needs topics

Once the worker's task-stage has been identified, task-need analyzer can discover the task-need topics. This work discovers a variety of task-need topics at each stage by the indicators of "*generality*" and "*specificity*" defined based on DO. *An ontology-based topic discovery method* is proposed to tackle the problem.

7.3.1 Determination task-needs on topics

There are two steps in discovering the worker's needed task-relevant topic, i.e., task-need topics. Step 1 identifies top task-relevant topics by similarity calculation between the task's temporal profile and the task topics in the DO. Step 2 determines the generality and specificity of task-need topics by examining the variety of topics within the DO. Notably, the leaf nodes in the DO, e.g. task-level topic nodes, represent the specific task topics, whereas the none-leaf nodes in the DO, e.g. field-level topic nodes, represent the general topics. The task-need topic may be a general or a specific task-relevant topic within the DO.

As we mentioned in Section 7.1, a set of top task-relevant topics (denoted as *TRTs*) is identified to model the worker's task-needs. The examination procedure in Step 2 is a top-down process. We first check the nodes at the field-level to examine the *generality* and *specificity* of task-needs, and then check the nodes at the task-level to examine the *specificity* of task-needs. The output of the discovery process expresses the worker's task-needs on topics with associated "*generality*" and "*specificity*" indicators, as shown in Figure 12.

Generality of task-need topics: We now calculate generality of task-needs topics within the transaction based on Eq. 7.3. As the DO in Figure 2 shows, a field-level topic may include one or more task-level topics. Therefore, the generality of a field-level topic, f_e, is the percentage of top *TRTs* in task-level to all task-nodes in f_e.

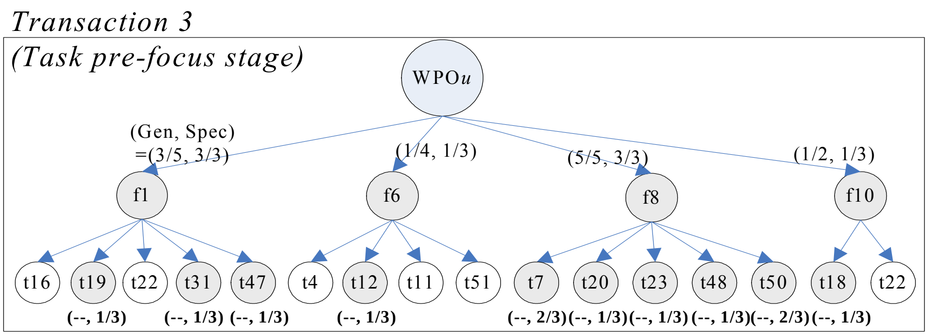


Fig. 12. Generality and specificity indicators

$$Gen(f_l)_{field-level} = \frac{N^{f_l}_{TRT}(Trans_i)}{N^{f_l}} \quad \textbf{(7.2)}$$

N^{f_l} denotes the number of task-level topics belonging to field l in the proposed DO. $N^{f_l}_{TRT}(Trans_i)$ is the number of distinct task-level topics belonging to field l and *TRT*s of transaction i.

Specificity of task-need topics: The equations below show the specificity of a topic f_l, in the field-level and the specificity of a topic t_k, in the task-level.

$$Spec(f_l)_{field-level} = \frac{\sum_{session\ j} B^{f_l}_{i,j}}{S_i} \quad \textbf{(7.3)}$$

$$Spec(t_k)_{task-level} = \frac{\sum_{session\ j} B^{t_k}_{i,j}}{S_i} \quad \textbf{(7.4)}$$

S_i is the number of sessions within a transaction i. $B^{f_l}_{i,j} = 1$ if f_l is a top relevant topic of session j in transaction i; otherwise 0. Similarly, $B^{t_k}_{i,j} = 1$ if t_k is a top relevant topic of session j in transaction i; otherwise 0. Notably, the system uses *TRTWs*(*Trans*) described in Section 7.1 to determine whether a topic (t_k or f_l) is a top relevant topic of transaction i (session j). The summation of $B^{t_k}_{i,j} / B^{f_l}_{i,j}$ counts the number of sessions in which the topic (t_k or f_l) is a top relevant topic, as shown in Table 15. The specificity of a topic (t_k or f_l) represents the ratio of top-relevance occurrences of the topic in the sessions of a transaction. For example, based on Eq. 7.3 and Eq. 7.4, the specificity of topic 8 in field-level is $Spec(f_8)_{field-level} = 3/3$ and the specificity of task 7 in task-level is $Spec(t_7)_{task-level} = 2/3$.

The "*generality*" and "*specificity*" indicators are used to determine the general and specific topics. That is, a topic with "*generality*" or "*specificity*" greater than a predefined threshold is regarded as a general or specific topic. Such kind of topic is a task-need topic at each stage.

Table 15. Summation of $B^{t_k}_{i,j} / B^{f_l}_{i,j}$ in TRTWs of *Trans3*

Task-level topics											**Field-level topics**			
Topics	**t7**	t12	t18	t19	t20	t23	t31	t47	t48	**t50**	**f01**	f06	**f08**	f10
Frequenc	**2**	1	1	1	1	1	1	1	1	**2**	**3**	1	**3**	1

Notes: Sets of TRTWs are listed in Example 2. (S_i=3)

7.4 Knowledge support based on task-stage

In this section, we explain how to conduct knowledge support based on the discovering result of a worker's task-stage and task-need topics at each stage. The profile adaptation considers the worker's task-stage and task-need topics of stages to adjust task profiles.

7.4.1 Profile adaptation

The new task profile of the target task, denoted as $\vec{S}_{p+1}$ is generated by the profile adaptation equation defined in Eq. 7.5. The equation considers the worker's task stage and the generality and specificity of task-need topics.

$$\vec{S}_{p+1} = \alpha\vec{S}_p + \lambda\,\vec{R} + (1-\lambda)\overrightarrow{Trans}_i$$
$$\vec{R} = w_{Gen}\sum\nolimits_{\forall topic_j \in Gen.\ topic} Gen(topic_j)\times w_{p+1}(topic_j)\,\overrightarrow{topics}_j + \qquad \textbf{(7.5)}$$
$$w_{Spec}\sum\nolimits_{\forall topic_j \in Spec.\ topic} Spec(topics_j)\times w_{p+1}(topic_j)\,\overrightarrow{topics}_j$$

where $\vec{S}_p$ denotes the task profile of the target task at time *p*. The relevant feature vector is derived from the task corpora of relevant tasks sets, and the temporal profile, $\overrightarrow{Trans}_i$. $w_{p+1}(topic_j)$ denotes the associated degree of relevance (i.e., rd_j is calculated in step 1 of Section 7.2.1) of $topic_j$ (t_k or f_l) to the target task. $Gen(topic_j)$ and $Spec(topic_j)$ are derived from the task-need topic analysis described in Section 7.3.
There are two factors to influence the profile adaptation equation. One is task stage and the other is task-need for specific topics at each stage. The parameter α is the correlation values between transactions, $corr_u(Patt_{Trans_{i-1}}, Patt^{S_1}_{Trans_i})$. The parameter λ is used to adjust the relative importance of relevant tasks and the temporal profile. As the task progresses, the content of the temporal profile becomes more important than that of the task-relevant topic. That is, the value of λ will decrease such that the influence of temporal profile increased as the task progresses. Meanwhile, the task-stage will also influence the relative weight of general topics, w_{Gen} , and specific topics, w_{Spec}. For example, in the early stage of a task, a worker tends to have general interesting in topics; therefore, the general topics are more important than specific topics. As the task progresses, the specific topics are more important than general topics. Note that w_{Gen}+ w_{Spe}.=1.

Table 16. Parameters adjustment across task-stages

	Parameters	Pre-focus	Focus	Post-focus
Task stage	α	$corr_u(Patt_{Trans_{i-1}}, Patt^{S_1}_{Trans_i})$		
	λ	Decreased the value across task-stage		
Task-need topic	w_{gene}	Decreased the value across task-stage		
	w_{spec}	Increased the value across task-stage		

7.4.2 Knowledge support

The generated task profile is the system kernel that streamlines knowledge retrieval activity to further realizes task-stage knowledge support. A task profile specifies key subjects of the executing task, and is constructed to model the information needs of knowledge workers based on the task-stage. Moreover, the task profile can be further adjusted based on the identification of task-stage by monitoring the workers' feedback behavior. The most task-relevant codified knowledge items can be retrieved to fit the worker's current task-needs.

Based on task profiles, the system can provide proactive delivery of task-relevant codified knowledge items from the repository to assist knowledge workers. The similarity measures between the task profile of executing task and the codified knowledge items can be calculated to select Top-N relevant tasks or documents from the knowledge repository. The key contents of a codified knowledge item (task or document) are represented as a feature vector of weighted terms. The task profile is also expressed as a feature vector of weighted terms. The cosine measure of feature vectors is used as similarity measure, which is given by Eq. 2.2.

7.5 Experimental setup

For evaluating the effectiveness of the proposed task-stage knowledge support model, we conduct empirical investigation in our problem domain. Section 7.5.1 reviews the experimental objective and procedure, respectively. Meanwhile, the remaining subsections describe the data set and participants, evaluation metrics and related parameter selection.

7.5.1 Overview of experiments

Experimental objective and procedure

Two experiments were performed to evaluate if the proposed task-need evolution discovery techniques can deliver task-relevant knowledge more precisely. Two techniques are proposed: one is the *task-stage identification technique*, and the other is the *ontology-based topic discovery technique*. Herein, the task-relevant knowledge means codified knowledge, i.e., textual data (research note, paper, etc.) accessed and created by tasks. Accordingly, the objectives of experimental evaluations were twofold: (1) Experiment one evaluates if providing knowledge support based on the determination of the worker's current task-stage can deliver task-relevant information; (2) Experiment two evaluates if discovering the worker's task-need topics of stages can learn the worker's task-needs more precisely. That is, the system can delivery more pertinent information to the workers with the aid of topic identification.

Table 17 lists the methods compared in this work. Two baseline methods are designed. Each proposed method is compared with the baseline methods. The baseline methods are designed based on the traditional incremental learning model in the information filtering (IF) system, as described in the literature review. In this work, we called the baseline method--*incremental learning technique,* since the system learns the users' current interests from the feedback on the recommended information (e.g., documents), and updates its model of the user for future information filtering. The two baseline learning methods have a slightly difference. The first is Learning-0 method that only considers a worker's feedback behavior on documents. The second is Learning-0.5 method that considers both a worker's feedback behavior on documents and task-relevant topics. That is, two kinds of information, i.e., feature set derived from documents and topic profiles, are equally important while conducting profile adaptation. The learning methods in *incremental learning technique* considered the worker's feedback behavior without considering the worker's current stage of task performance. However, in this work, the changes of worker's task stage and the worker's task-needs topic are both incorporated into the traditional IF model. Thus, the system considers the worker's feedback, task stage and task-needs topic to provide a more elaborative information filtering.

Table 17. Experiments description

Technique	Method	*Description*	Parameter Setting
Task-stage identification technique	❷ Stage Method	Document support considered worker's task-stage, as described in Section 7.2.	Stage Method with $\alpha = 1,\ \lambda = f_{stage}(task\_stage_i)$ $\vec{R} = \sum_{\forall t_j \in TRTs(Trans_i)} w_p(topic_j)\ \overrightarrow{topics}_j$
	❸ Stage-C Method	Document support considered worker's task-stage and correlation value between transactions, as described in Section 7.2.	Stage-C Method with $\alpha = corr_u(Patt_{Trans_p}, Patt^{S_1}_{Trans_{p+1}}),\ \lambda = f_{stage}(task\_stage_i)$ $\vec{R} = \sum_{\forall t_j \in TRTs(Trans_i)} w_p(topic_j)\ \overrightarrow{topics}_j$
Ontology-based topic discovery technique	❹ Stage-C-Topic Method	Document support considered a worker's general and specific task-needs topic at each task-stage, as described in Section 7.3.	Stage-Topic Method with $\alpha = corr_u(Patt_{Trans_p}, Patt^{S_1}_{Trans_{p+1}}),\ \lambda = f_{stage}(task\_stage_i)$ $\vec{R} = \sum_{\forall t_j \in Gene.\ topic} Gene(topic_j) \times w_p(topic_j)\ \overrightarrow{topics}_j + \sum_{\forall t_j \in Spec.\ topic} Spec(topics_j) \times w_p(topic_j)\ \overrightarrow{topics}_j$
Incremental learning technique	❶ Similar to standard Rocchio algorithm in relevance feedback	Baseline (Learning worker's information need mainly considered a worker's feedback behavior on documents)	Linear-0 Method (or Linear-0.5 Method) with $\alpha = 1,\ \lambda = 0\ (\text{or } 0.5)$ $\vec{R} = \sum_{\forall t_j \in TRTs(Trans_i)} w_p(topic_j)\ \overrightarrow{topics}_j$

Experiment one compares the task-stage identification technique with the baseline technique. Two methods are designed based on the *stage identification technique*, as described in Section 7.2. One is Stage method (denoted as Stage method), and the other is Stage-Correlation (denoted as Stage-C method) method. The Stage method considers the worker's current task stage to adjust the relative importance of task relevant topics and the temporal profile. Thus, the parameter λ is adjusted based on the worker's current task stage, as shown in Table 16. The Stage-C model is also built upon the Stage method. Besides, the correlation value of the worker's task-need patterns across transactions is incoporated into the model. Namely, the parameter α is setting to the value of the correlation value between transactions instead of setting to 1 in traditional relevance feedback method. Consequently, this experiment evaluates the effectiveness of Stage method, and Stage-C method versus incremental learning method.

Experiment two is an extension of experiment one. One method is designed based on the proposed *ontology-based topic discovery technique*. The method is denoted as Stage-C-Topic method. The Stage-C-Topic method discovers the worker's task-needs for specific topics according to the indicator of generality and specificity. The topic's generality, i.e., *Gen(topic)*, or specificity value, i.e., *Spec(topic)* above the threshold will be considered as task-need topic at the corresponding stage. Thus, the topic profile (feature vector of weighted terms) with the associated value of *Gen(topic)* or *Spec(topic)* will influence the proposed profile adaptation equation. Meanwhile, the parameter λ is also adjusted based on the worker's current task stage, and α is set to the value of the correlation value between transactions, as shown in Table 16. Consequently, this experiment evaluates the effectiveness of Stage-C-Topic method versus Stage-C method.

7.5.2 Data, participants and evaluation metrics

Experiments were conducted using a real application domain on conducting research tasks in a laboratory of a research institute. Note that knowledge workers usually require a longer time (e.g. one year) to accomplish knowledge-intensive tasks, namely the whole process of task performance spans across a long time period. The real application domain and the characteristic of the knowledge-intensive task may restrict the sample size of the data and participants in the experiments.

Data and Participants

Task and Participants: In this work, the tasks concerned are writing research papers or conducting research projects. Fifty research tasks were collected, with 31 existing tasks, and 19 executing tasks. We randomly selected evaluation subjects who are engaged in the executing-task. As we have mentioned, the whole process of task performance spans across a long time period. Thus, we choose evaluators according to their progress of task execution, i.e., pre-focus, focus formulation, or post-focus task stages, for avoiding evaluators are all within the same task stage. Specifically, for evaluating the effectiveness of the task-stage knowledge support model, we selected 5 subjects at a specific stage from the existing-task set as the evaluation subjects, namely fifteen evaluations totally.

Data: We examine the support effectiveness by examining the effectiveness of the retrieval result at different task-stage. The target set is the documents in the organizational knowledge repository that have been accessed or generated from the existing-task set, i.e., historical task. Over 600 documents are collected during the period of 2002~2005.

Performance evaluation metrics

The effectiveness is measureed in terms of precision, recall and F1-measure, as in information retrieval.

Precision and recall. *Precision* is the fraction of retrieved items (tasks or documents) that are relevant, while *recall* is the fraction of total known relevant items that are retrieved. The definition is given in Eq. 5.3 and Eq. 5.4 of Section 5.5.2.

F-Measure. For observing the relative importance of precision and recall, we use a combination metric F-Measure to adjust the relative weight of precision and recall. The F1-metric [61][62] could be used to balance the trade-off between precision and recall.

$$F_{\beta} = \frac{(1+\beta^2)\times precision \times recall}{\beta^2 \times prcision + recall} \tag{7.6}$$

The value of β is to adjust the relative importance of the recall in comparison to the precision. If β=0, F_{β} coincides with precision, and if $\beta = \infty$, F_{β} coincides with recall. In this experiment, we set β=1 (Recall and precision are equally important).

7.6 Experimental results and implications

7.6.1 Experiment one: effect on task-stage identification

Knowledge support: Observations by stages

Figure 13 shows the performance of four methods in terms of precision, recall, and F-measure by averaging three stages. In addition, Table 18 shows the result of knowledge support for document-retrieval based on three stages. Four models are evaluated, including Liner-0 method, Linear-0.5 method, Stage method, and Stage-C method. The Linear-0 method, and Linear-0.5 method are our baseline methods. The Stage method, and Stage-C method are the proposed methods based on the *task-stage identification technique.*

Observation 1: Figure 13 shows that the average value of precision, recall, and F-measure of Stage-C method over three stages exceeds those of the other methods. The experimental result reveals that building task profile by the proposed *task-stage identification technique* can retrieve more task-relevant documents to knowledge workers.

Observation 2: Look more details, Table 18 shows that the Stage-C method can achieve better performance than the other three methods, especially in stage one and two. The result indicates that the worker's may have high variations on topics in the early stage of task performance, i.e., uncertainty about the research topic; therefore,

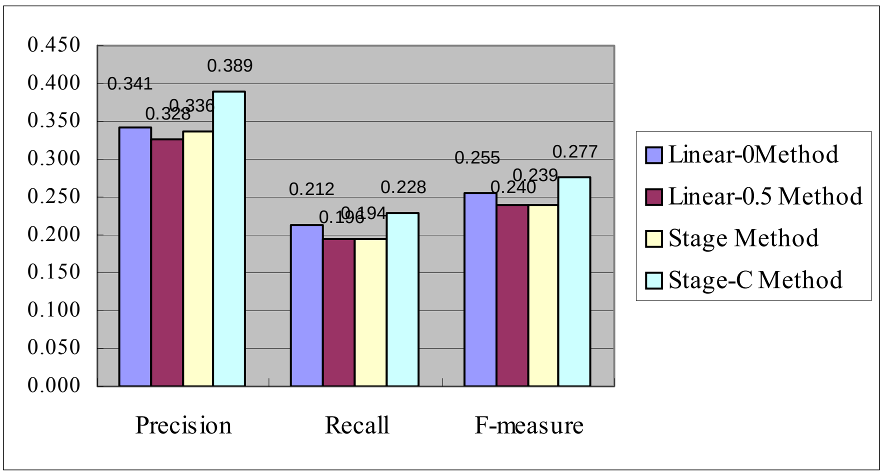


Fig. 13. Result of knowledge support by averaging stages (performance value in *y* axes)

Table 18. Result of knowledge support by stages (top-30 document support)

Stage		Linear-0 Method			Linear-0.5 Method			Stage Method			Stage-C Method		
***n*th Stage**	**Case**	**Pre.**	**Re.**	**F-measure**	**Pre.**	**Re.**	**F-measure**	**Pre.**	**Re.**	**F-measure**	**Pre.**	**Re.**	**F-measur**
1st stage (pre-focus)	**1**	0.433	0.351	0.388	0.400	0.324	0.358	0.467	0.378	0.418	0.533	0.432	0.477
	2	0.113	0.118	0.115	0.113	0.118	0.115	0.233	0.206	0.219	0.233	0.206	0.219
	3	0.267	0.195	0.225	0.300	0.220	0.254	0.200	0.025	0.044	0.400	0.049	0.087
	4	0.267	0.136	0.180	0.300	0.153	0.203	0.333	0.169	0.224	0.333	0.169	0.224
	5	0.100	0.051	0.068	0.167	0.085	0.113	0.167	0.085	0.113	0.167	0.085	0.113
Average		**0.236**	**0.170**	**0.195**	**0.256**	**0.180**	**0.209**	**0.280**	**0.173**	**0.204**	**0.333**	**0.188**	**0.224**
2nd stage (Focus formulation)	**6**	0.133	0.125	0.129	0.167	0.156	0.161	0.167	0.156	0.161	0.233	0.219	0.226
	7	0.200	0.176	0.187	0.200	0.176	0.187	0.233	0.206	0.219	0.333	0.294	0.312
	8	0.633	0.196	0.299	0.700	0.216	0.330	0.633	0.196	0.299	0.633	0.196	0.299
	9	0.667	0.408	0.506	0.600	0.367	0.455	0.667	0.408	0.506	0.667	0.408	0.506
	10	0.233	0.135	0.171	0.267	0.154	0.195	0.267	0.154	0.195	0.333	0.192	0.244
Average		**0.373**	**0.208**	**0.259**	**0.387**	**0.214**	**0.266**	**0.393**	**0.224**	**0.276**	**0.440**	**0.262**	**0.317**
3rd stage (Post-focus)	**11**	0.233	0.156	0.187	0.233	0.156	0.187	0.267	0.178	0.214	0.367	0.244	0.293
	12	0.467	0.233	0.311	0.400	0.200	0.267	0.400	0.200	0.267	0.400	0.200	0.267
	13	0.400	0.200	0.267	0.367	0.183	0.244	0.367	0.183	0.244	0.400	0.200	0.267
	14	0.567	0.298	0.391	0.567	0.298	0.391	0.567	0.298	0.391	0.567	0.298	0.391
	15	0.400	0.400	0.400	0.133	0.133	0.133	0.067	0.067	0.067	0.233	0.233	0.233
Average		**0.413**	**0.257**	**0.311**	**0.340**	**0.194**	**0.244**	**0.334**	**0.185**	**0.236**	**0.393**	**0.235**	**0.290**

the profile adaptation considered the worker's task stage can conduct better knowledge support. For example, if a worker is the early task stage, he/she may have broad task-needs on topics and may change topics sometimes. Thus, the correlation value, $\alpha = corr_u(Patt_{Trans_p}, Patt^{S_1}_{Trans_{p+1}})$, between transactions is negative. The feature set in task profile of previous transaction will be subtracted from the feature set in task profile of current transaction. Thereby, irrelevant feature terms will be removed from the profile. However, the effect of changes of topics cannot be reflected in the incremental learning techniques.

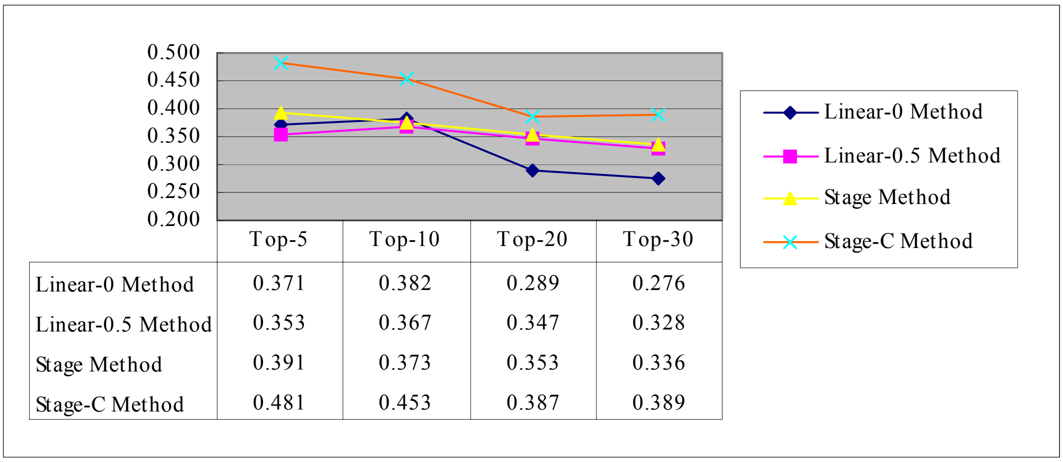
Observation 3: On the other hand, we observed that the incremental learning technique, Liner-0 method, has better performance than the other three methods in the third stage, post-focus stage. If we took a further analysis in each case in the third stage, we found 2 of 5 cases have better performance value by Liner-0 method. That is, since the worker has focused on a specific topic, it is more reliable to learn the worker's task-needs by feedback analysis. Thus, the stage effect will decrease, and the Liner-0 method has better performance than the stage methods.

Knowledge support: Observations under various top-N

Figure 14 shows the performance of four methods in terms of precision, and recall by averaging three stages. In addition, Table 19 shows the result of task-relevant document support based on three stages under various top-N. Four methods are also evaluated, including Liner-0 method, Linear-0.5 method, Stage method, and Stage-C method. Note that we focus on precision and recall values in this part since we would like to see the precision under various recall level.

Observation 1: Figure 14 shows the average precision values of each case at each stage under top-5, 10, 20 and 30. The result reveals that the Stage-C method can achieve better performance than other three methods, especially for top-5 document support. The result indicates the Stage-C method can provide more effective knowledge support than the other methods (the higher precision at the lower recall value).

Observation 2: Table 19 shows that the average value of precision and recall of Stage-C method under various top-N, i.e., top-5, 10, 20 and 30, exceeds those of the other methods. The experimental result reveals that no matter the number of supporting document set, the Stage-C method can retrieve more task-relevant documents than the other methods.



	Top-5	Top-10	Top-20	Top-30
Linear-0 Method	0.371	0.382	0.289	0.276
Linear-0.5 Method	0.353	0.367	0.347	0.328
Stage Method	0.391	0.373	0.353	0.336
Stage-C Method	0.481	0.453	0.387	0.389

Fig. 14. Result of knowledge support by averaging stages under various top-N (Precision value in *y*-axes)

Observation 3: Particularly, we found a common situation exists in general: the fewer supported documents, the higher precision value. Thus, the system is suitable to be adopted in a working environment in which workers has time pressure to find task-relevant documents without taking times to review many documents. That is, it implies we could build an interactive knowledge support system for providing relevant documents of few retrieved documents and further to refine the task profile by feedback mechanism.

Implications: The overall experimental results demonstrate the proposed knowledge support model considering worker's task-stage is effective. Apparently, the Stage-C method has the best performance in each evaluation case. Therefore, the result reveals that take into account the correlation value between transactions is required for improving the effectiveness of knowledge support. Therefore, further experiments for evaluating ontology-based topic discovery technique will be tested under the modification of Stage-C method.

Table 19. Result of knowledge support by stages under various top-N (Experimental one)

Stage		Linear-0 Method		Linear-0.5 Method		Stage Method		Stage-C Method	
***n*th Stage**	**Top-N**	**Pre.**	**Re.**	**Pre.**	**Re.**	**Pre.**	**Re.**	**Pre.**	**Re.**
1st stage (Pre-focus)	**Top-5**	0.360	0.043	0.280	0.033	**0.373**	**0.077**	0.353	0.071
	Top-10	0.300	0.074	0.340	0.079	0.320	0.082	**0.380**	**0.099**
	Top-20	0.240	0.116	0.280	0.129	0.260	0.111	**0.310**	**0.149**
	Top-30	0.236	0.170	0.256	0.180	0.280	0.173	**0.333**	**0.188**
2nd stage (Focus formulation)	**Top-5**	0.360	0.028	0.540	0.046	0.520	0.105	**0.600**	**0.058**
	Top-10	0.420	0.071	0.480	0.091	0.460	0.083	**0.500**	**0.090**
	Top-20	0.417	0.170	0.430	0.158	0.410	0.130	**0.440**	**0.165**
	Top-30	0.373	0.208	0.387	0.214	0.393	0.224	**0.440**	**0.262**
3rd stage (Post-focus)	**Top-5**	0.393	0.158	0.240	0.023	0.280	0.026	**0.490**	**0.043**
	Top-10	0.426	0.120	0.280	0.055	0.340	0.063	**0.480**	**0.096**
	Top-20	0.210	0.077	0.330	0.099	0.390	0.148	**0.410**	**0.181**
	Top-30	0.220	0.118	0.340	0.194	0.334	0.185	**0.393**	**0.235**
Total Average		0.336	0.111	0.367	0.114	0.373	0.121	**0.422**	**0.136**

7.6.2 Experiment two: effect on discovery of task-needs topics

This experiment aims to determine the worker's task-needs topics, i.e., general and specific topics at each task stage. The proposed ontology-based topic discovery technique, named Stage-C-Topic method, is employed to analyze the general and specific task-relevant topics at each stage. Accordingly, the "generality" and "specificity" indicators are used to determine the general and specific topics, denoted as $Gen(topic_j)$ and $Spec(topic_j)$, respectively. Only topics whose value of $Gen(topic_j)$ or $Spec(topic_j)$ above the threshold are considered as task-needs for specific topics. Thus, the $Gen(topic_j)$ or $Spec(topic_j)$ value with associated topic profile, $\overrightarrow{topic_j}$ will be incorporated into the proposed profile adaptation equation. The Stage-C-Topic method considered the $Gen(topic_j)$ or $Spec(topic_j)$ value with associated topic profile. Note that the experiment is an extension experiment of experiment one. Therefore, the worker's task-stage is also considered in the proposed topic discovery technique. Table 17 enumerates the related parameters in the experiments in details. We evaluate and compare the performance in precision, recall, and F-measure of Stage-C method, and Stage-C-Topic method.

Knowledge support: Observations by stages

Table 20 shows the result of knowledge support for document-retrieval based on three stages. In addition, Figure 15 shows the performance of four methods in terms of precision, recall, and F-measure by averaging three stages. Three models are evaluated, including Liner-0 method, Stage-C method, and Stage-C-Topics method. The Linear-0 method, and the Stage-C method are used as the reference method, which has been evaluated in experiment one. The Stage-C-Topic method is the proposed method based on the *ontology-based topic discovery technique.* Note that the Stage-C-Topic method is modified based on the Stage-C method since the Stage-C method has the best performance in experiment one.

Observation 1: Figure 15 shows the performance of three methods by averaging the values of three stages. The result shows that Stage-C-Topic has higher performance (precision) value than the other methods. The result reveals that knowledge support by identifying task-relevant topics based on task-stages can achieve better performance.

Observation 2: Table 20 shows that the Stage-C-Topic method can achieve better performance than the other three methods, especially knowledge support in stage two and three. Although, Stage-C method is slightly better than Stage-C-Topics method in the first stage. But, if we took a further look to analyze each case in stage one, we found there is only one case by Stage-C method that has better performance than that of Stage-C-Topic method. That is, Stage-C-Topic method still can ensure better knowledge support on average.

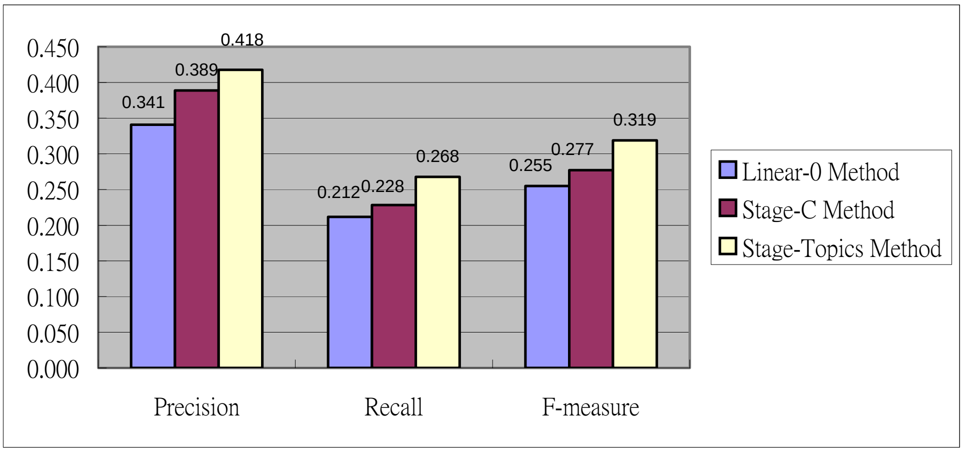


Fig. 15. Result of knowledge support by averaging stages

Table 20. Result of knowledge support by stages (experimental two)

Stage		Linear-0 Method			Stage-C Method			Stage-C-Topics Method		
***n*th Stage**	**Case**	**Pre.**	**Re.**	**F-measure**	**Pre.**	**Re.**	**F-measure**	**Pre.**	**Re.**	**F-measure**
1st stage (pre-focus)	**1**	0.433	0.351	0.388	0.533	0.432	0.477	0.567	0.460	0.508
	2	0.113	0.118	0.115	0.233	0.206	0.219	0.267	0.235	0.250
	3	0.267	0.195	0.225	0.400	0.049	0.087	0.267	0.195	0.225
	4	0.267	0.136	0.180	0.333	0.169	0.224	0.333	0.169	0.224
	5	0.100	0.051	0.068	0.167	0.085	0.113	0.200	0.102	0.135
Average		**0.236**	**0.170**	**0.195**	**0.333**	**0.188**	**0.224**	**0.327**	**0.232**	**0.269**
2nd stage (Focus formulation)	**6**	0.133	0.125	0.129	0.233	0.219	0.226	0.233	0.219	0.226
	7	0.200	0.176	0.187	0.333	0.294	0.312	0.333	0.294	0.312
	8	0.633	0.196	0.299	0.633	0.196	0.299	0.633	0.196	0.299
	9	0.667	0.408	0.506	0.667	0.408	0.506	0.667	0.408	0.506
	10	0.233	0.135	0.171	0.333	0.192	0.244	0.433	0.250	0.317
Average		**0.373**	**0.208**	**0.259**	**0.440**	**0.262**	**0.317**	**0.460**	**0.273**	**0.332**
3rd stage (Post-focus)	**11**	0.233	0.156	0.187	0.367	0.244	0.293	0.367	0.244	0.293
	12	0.467	0.233	0.311	0.400	0.200	0.267	0.500	0.250	0.333
	13	0.400	0.200	0.267	0.400	0.200	0.267	0.433	0.216	0.288
	14	0.567	0.298	0.391	0.567	0.298	0.391	0.533	0.280	0.367
	15	0.400	0.400	0.400	0.233	0.233	0.233	0.500	0.500	0.500
Average		**0.413**	**0.257**	**0.311**	**0.393**	**0.235**	**0.290**	**0.467**	**0.298**	**0.356**

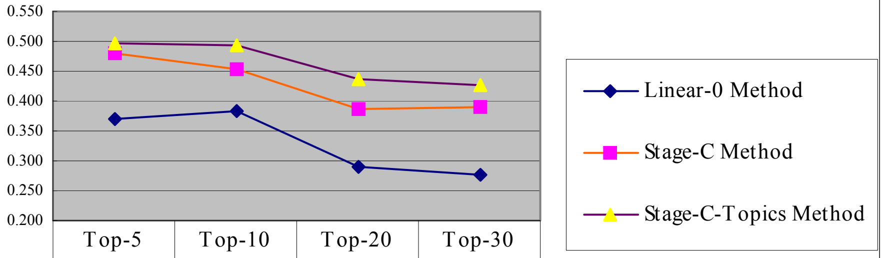
Knowledge support: Observations under various top-N

Figure 16 shows the performance of four methods in terms of precision, recall, and F-measure by averaging three stages under various top-N values. Note that we also focus on precision and recall values in this part since we would like to see the precision under various recall level.

Observation 1: By average, the precision values of Stage-C-Topics method under various top-N, i.e., top-5, 10, 20 and 30, of document support exceed those of the other methods. The experimental result reveals that no matter the number of supporting document set, the Stage-C-Topic method can retrieve more task-relevant documents than the other methods. Note that, Stage-C method and Stage-C-Topic method have similar precision under top-5 document supports. That is, stage-C-Topic method may have more significant effect while supporting more documents.

Observation 2: If we took a further look at Table 21, we found that the performance values of Stage-C-Topic method under various top-N document supports at each task stage are better than those of the other methods. This result indicates Stage-C-Topic can ensure the quality of knowledge support under various number of document support. In addition, task-stage topic discovery, which identifies general and specific topics, has positive effect while conducting knowledge support.

Implications: The result reveals that task-stage knowledge support considering both worker's task-stage and task-need topics at the same time can ensure the better performance of knowledge support than that of only considering worker's task-stage (Stage-C method). We are encouraged from the result of Stage-C-Topic method in supporting more number of task-relevant documents. In further work, we will improve the technique from two aspects: (1) we may test various threshold values to filter the specific and general task-need topics; and (2) we may test the various combinations of relative importance of general and specific topics. That is, we will consider the impact of task-stage for general topics and specific topics, since most studies reveals that as the task progresses, the worker will dedicate to specific topics. Accordingly, the task-stage will also influence the relative importance of general and specific topics.



	Top-5	Top-10	Top-20	Top-30
Linear-0 Method	0.371	0.382	0.289	0.276
Stage-C Method	0.481	0.453	0.387	0.389
Stage-C-Topics Method	0.498	0.493	0.437	0.427

Fig. 16. Result of knowledge support by averaging stages under various top-N (Precision value in *y*-axes)

Table 21. Result of knowledge support by stages under various top-N (Experimental two)

Stage		**Linear-0 Method**		**Stage-C Method**		**Stage-C-Topics Method**	
***n*th Stage**	**Top-N**	**Pre.**	**Re.**	**Pre.**	**Re.**	**Pre.**	**Re.**
1st stage (Pre-focus)	**Top-5**	0.360	0.043	0.353	0.071	**0.373**	**0.078**
	Top-10	0.300	0.074	0.380	0.099	**0.380**	**0.105**
	Top-20	0.240	0.116	0.310	0.149	**0.330**	**0.143**
	Top-30	0.236	0.170	0.333	0.188	**0.353**	**0.203**
2nd stage (Focus formulation)	**Top-5**	0.360	0.028	**0.600**	**0.058**	0.560	0.052
	Top-10	0.420	0.071	0.500	0.090	**0.580**	**0.108**
	Top-20	0.417	0.170	0.440	0.165	**0.490**	**0.189**
	Top-30	0.373	0.208	0.440	0.262	**0.460**	**0.273**
3rd stage (Post-focus)	**Top-5**	0.393	0.158	0.490	0.043	**0.560**	**0.058**
	Top-10	0.426	0.120	0.480	0.096	**0.520**	**0.112**
	Top-20	0.210	0.077	0.410	0.181	**0.490**	**0.211**
	Top-30	0.220	0.118	0.393	0.235	**0.467**	**0.298**
Total Average		0.336	0.111	0.422	0.136	**0.447**	**0.146**

Chapter 8 $\mathcal{K}$-Support System

8.1 System architecture

Figure 17 depicts the system architecture comprising four implementation layers, including knowledge resource collection, knowledge acquisition, knowledge modeling, and Web-based front-end application.

Knowledge resource collection layer: The unstructured or semi-structured information embedded in records such as documents, presentation slides, reports, lesson-learned, database entries, etc., are valuable knowledge items. This layer collects information expressed in various forms from different knowledge sources that are generated and accessed during task executions. Meanwhile, the system collects data from human resource applications to provide a platform for gathering and exchanging task-relevant knowledge among workers.

Knowledge acquisition layer: This layer extracts explicit (codified) knowledge and tacit (human resource) knowledge within the organization. Two modules are responsible to handle and process task-relevant knowledge items: one is the *data-processing module* and the other is the *task-processing module.* This layer employs information retrieval, text mining and database techniques to process and organize task-relevant information.

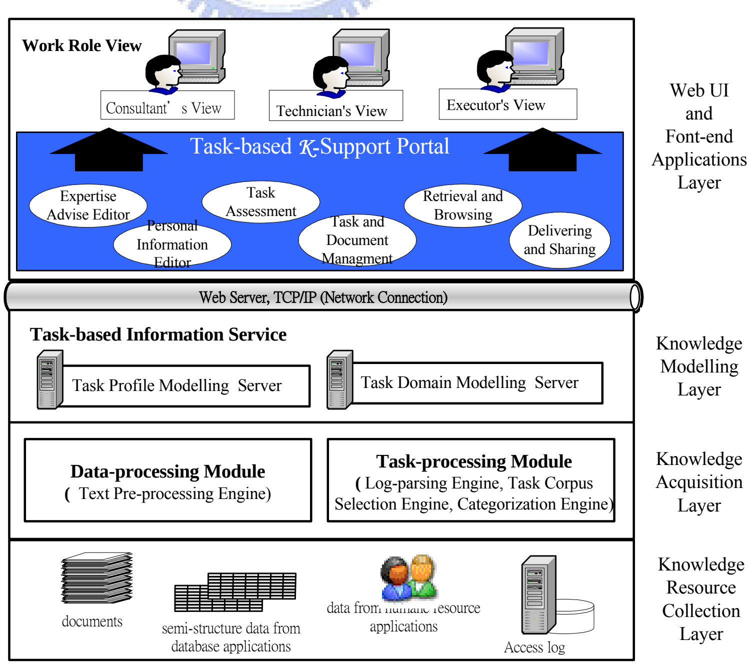


Fig 17. System architecture

- The *data-processing module* deals with textual data represented in different formats. The information extraction engine retrieves meaningful information such as title, abstract, and author name from documents. The text pre-processing engine employs *term transformation*, *term weighting,* and *feature selection* steps [7][60][67] to extract meaningful information (metadata) of textual-based knowledge items.

- The *task-processing module* comprises three processing units, including log-parsing, task corpus selection, and task categorization engines to handle task-relevant data. The log-parsing engine analyzes log-files to track user's interaction with the system. The task corpus selection engine generates the task corpus of a task t_r by analyzing the contents of textual data accessed by t_r. The task corpus represents the key features of a task. The task categorization engine is responsible for ontology configuration via a seed-based fuzzy classification technique.

Knowledge modeling layer: This layer is responsible for modeling task-relevant information and workers' information needs. The *domain modeling module* is responsible for representing domain ontology from the aspect of task. The module handles the grouping of similar tasks into fields. The ontology is used to represent the organization's domain-specific knowledge. Task related information (e.g. skills, knowledge, workers, and documents, etc.) is also conceptualized into the agreed ontology to provide knowledge support. The *profile modeling module* provides mechanisms such as profile creation, modification, and integration to conduct profile management. Moreover, the module implements the profile handler described in Section 3.2. Profile modeling is the kernel to support knowledge retrieval and sharing.

Web-based GUI and front-end applications layer: An integrated platform is built upon the *profile modeling server* and *domain modeling server* to construct the task-based knowledge support portal. This layer mainly provides the function of a task-oriented retrieval router described in Section 3.2. Moreover, the proposed system considers the task perspective to acquire and disseminate task-relevant knowledge. Different knowledge management applications are available for workers. For example, the function of *task assessment editor* assists a worker to conduct task assessment to create his/her own task profile. A worker may use the *personal* *information editor* to organize his/her own knowledge. In addition, a worker can enter the *task-based workspace* to browse, access, and organize task-relevant knowledge. The proposed system not only delivers task-relevant knowledge based on task profile but also identifies peer-groups with similar task-needs (or similar projects) based on work profiles. Workers engaged in the same task or with common task needs can solve the encountered problem together, thereby realizing collaborative task-based knowledge support.

8.2 System demonstration and scenario descriptions

The task-based *K*-Support portal is a Web-based application, allowing workers to retrieve, organize and share task-relevant knowledge. The K-Processing application is dealing with the operation of task-relevant objects. In addition, there are three main applications provided in the proposed portal. K-Assessment application assists an executor to conduct task assessment to create his/her own task profile. K-Delivery application, which delivers task relevant knowledge proactively to support task execution. K-Sharing application, which stimulates knowledge sharing by locating possible task-based peer-groups.

8.2.1 K-Processing: Task relevant information processing

The K-Processing mainly handle task-relevant information items. That is, the operation executed by *data-processing module* and *task-processing module* within the knowledge acquisition layer can be processed in this layer. Thus, a task-relevant expert can add, update, or delete an object, such as a worker object, a task object, a document object and so on via the K-Processing Interface. Meanwhile, the data-processing module, task processing module, or the categorization module can be activated through the interface. The standard Information Retrieval (IR) technology is employed to accomplish text processing. The Relational Database Management technique and Text Mining techniques are applied in both modules to process and organize textual data and task-relevant information. Figure 18 shows the interface of the K-Processing. Implementation details are given in our previous work [86].

8.2.2 K-Assessment: Identifying task-relevant knowledge

Scenario Description: The worker can obtain help from experts in conducting the assessment. The worker can access a list of referring tasks. He can arbitrarily click for further information about a specific task. Finally, after finishing the assessment, he can enter his task workspace to access the relevant knowledge sources provided by the system.

Operations via Interface: The worker can conduct task-assessment to generate his own task profile. If he selects the "assessment" item, the system will guide him to conduct two-phase task-relevance assessment. The worker should give his perceptions of each category. Besides the worker's perception about the task, he can choose the "expert" column to help him conduct assessment. The results of assessment are submitted to the system's task profile modeling server to compute the initial task profile. The task profile is expressed as a feature vector of weighted terms. Figure 19 shows the interface of the two-phase assessment procedure. The detail of collaborative task-relevance assessment is given in Chapter 5.

8.2.3 K-Delivery: Delivering codified knowledge proactively

Scenario Description: Everyone who finished the assessment can enter his task workspace. The system will recommend the task-relevant and the latest information based on the task profile Workers could accept or reject these knowledge items by clicking the feedback web form. Meanwhile, the system will receive feedbacks and modify worker's profile.

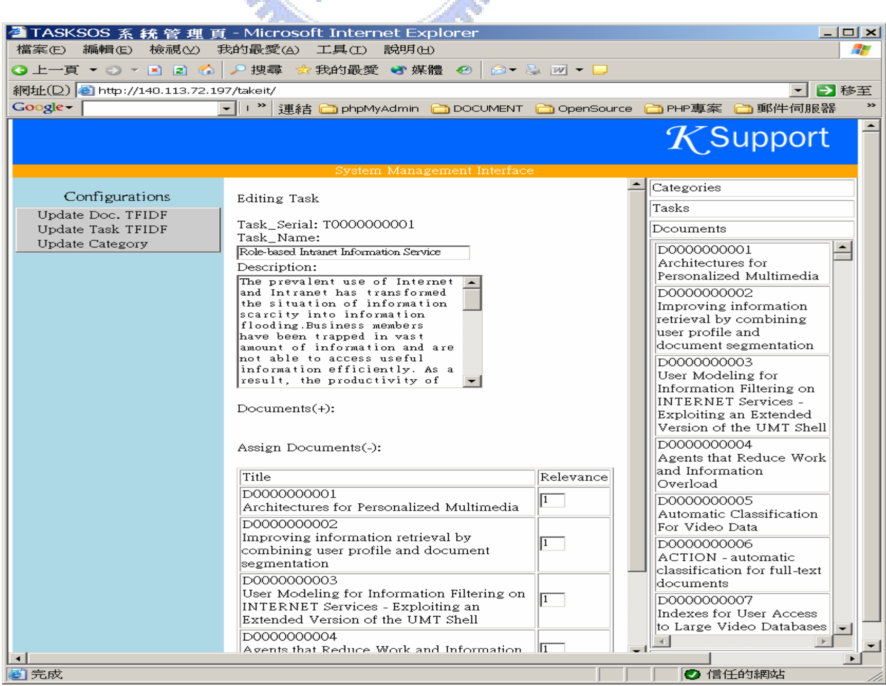


Fig. 18. Interface of K-Processing

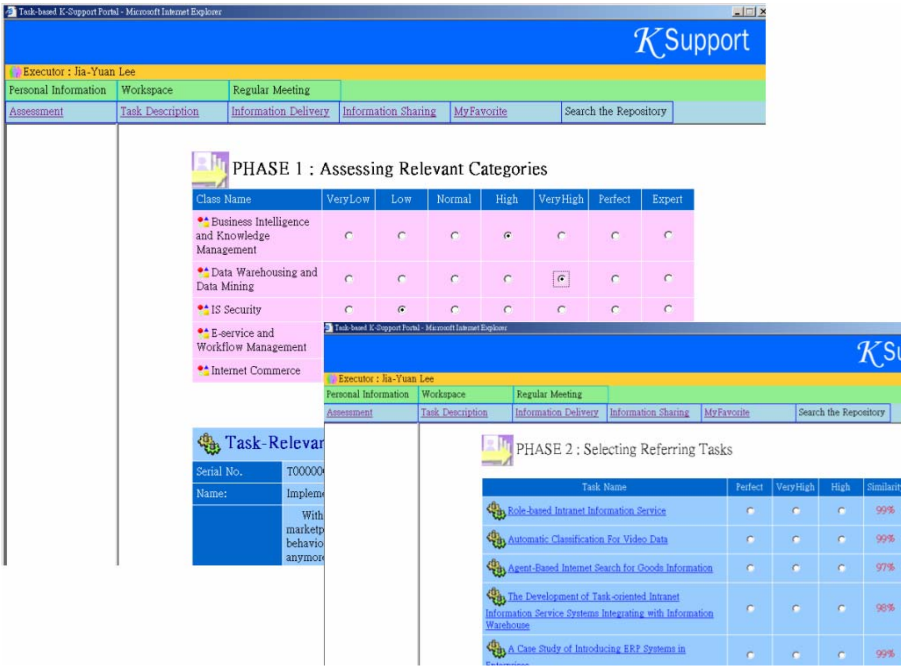


Fig. 19. Interface of two-phase assessment

Operations via Interface: The system can proactively deliver task-relevant information based on the worker's task profiles. Figure 20 shows the top-5 relevant tasks, top-30 relevant documents and 10 task-associated terms provided by the system. A tree-like structure is employed to organize task-relevant information. Once the worker selects a document or a task to read, the detailed information will be displayed, as shown in the right frame of Figure 20. Meanwhile, the worker can view the description of any task-relevant document, as denoted in circle 1. The worker can also conduct feedback on the recommended items. Six relevant degrees are provided by the system- "*very low*", "*low*", "*normal*", "*high*", "*very high*", and "*perfect*", as shown in Figure 21. If the worker gave a positive rating on the knowledge item (document or task sets), the system will preserve the item in the worker's MyFavorite folder. The detail of disseminating task-relevant knowledge is given in Chapter 6.

8.2.4 K-Sharing: Knowledge support from peer-group

Scenario Description: Once the worker cannot obtain knowledge support from the application of knowledge delivery, he/she can seek the assistance from the application of knowledge sharing. That is, the system identifies peer-groups with similar task needs based on work profiles. The system facilitates knowledge sharing by displaying the shared information such as relevant tasks and documents retrieved

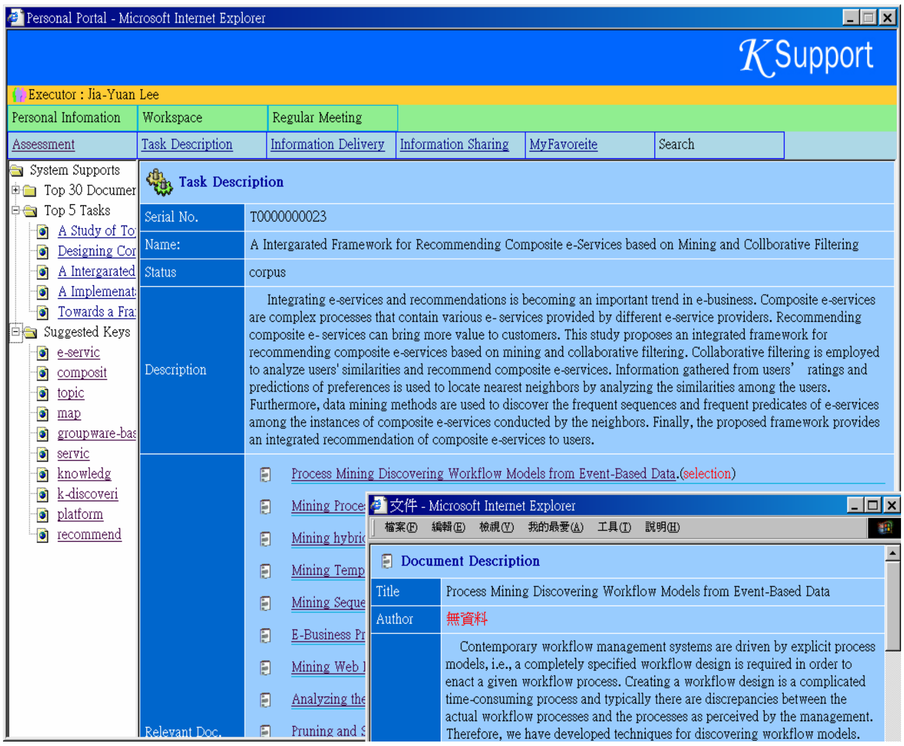


Fig. 20. Interface of knowledge delivery

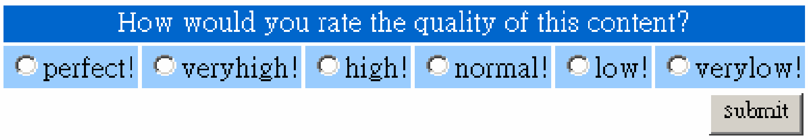


Fig. 21. Six Degrees of relevance feedback

from peer-group members. Note that all information is calculated timely and automatically according to the feedback results.

Operations via Interface: The system expands the personalized ontology of a worker with the peer-group member's personalized ontology for knowledge sharing. Notably, a worker's personalized ontology represents a worker's perspective of task-needs on the target task. The personalized ontology is derived from the work profile to record tasks or fields that are relevant to the target task. The left frame of Figure 9 in Section 6.4 shows the sharing tree of "Jia-Yuan Lee", as denoted in circle 1. A sharing tree is a tree-like structure, which represents the personalized ontology of a worker. Meanwhile, the shared information from task-based peer-groups is also presented in the sharing tree. In the given example, the ontology {H3.3 Information Retrieval and K4.3 Organization Impact, Mining Association Rule for Information Recommendation in Enterprises} is shared from "Mike Lee", as denoted in circle 2 of Figure 9. Another tree-like structure below the sharing tree is used to organize the shared document sets from the task-based peer-group (as denoted in circle 3). Notably, a threshold, α-cut level, which is shown in the top left frame, can be adjusted by the workers to find more peer-group members by decreasing the α value. The detail of disseminating and sharing task-relevant knowledge is given in Chapter 6.

8.3 Discussions

We also examined the user effort to conduct assessment procedure and the overall useful perception about the proposed system briefly more details are given in the previous publication [86].

User effort: The user effort result showed that the novice workers took 27 minutes on average to complete the procedure, whereas the experienced workers took 16 minutes on average. This result is in line with the research of Marshall and Byrd (1998) that states the use and perception of information system (IS) varies by user groups due to the task domain knowledge. In our post questionnaires, we found that experienced workers seemed satisfied with the design of assessment procedure, whereas the novice users seemed to take more time to conduct the assessment. Therefore, the collaborative mechanism is especially required for novices.

Satisfaction: For measuring the users' satisfaction of using the system, a 6-point Likert scale from 1 to 6 was deployed. Two questions were asked after completing the task assessment: one was the difficulty to conduct the task assessment and the other was the usefulness of the recommend items. Table 21 shows the result of the users' evaluation. Notably, the common consensus about usefulness among experienced workers was low measured by standard deviation (e.g., the value is higher than that of novices). This resulted from the fact that some of experienced workers were not satisfied with the quantity of knowledge support. Increasing system scalability is a major task in our future work.

Finally, according to our post questionnaires about the proposed system, the novices reflect that they want to get more help from humane resource, whereas the experienced workers need more number of task-relevant knowledge supports. It indicates that the sharing mechanism is more essential for novices than that of experienced workers. On the other hand, the web mining technique is more desirable for experienced workers for acquiring more task-relevant knowledge form the Web pages.

Table 21. Average of likert scale value form system evaluation (Higher is better, range=1-6)

	Novices		**Experienced Workers**	
	Average	**Std. deviation**	**Average**	**Std. deviation**
Easy to completed the task assessment	3.800	1.095	5.200	1.095
Useful to support task	4.200	0.448	4.400	1.342

Chapter 9 Conclusions and Future Works

9.1 Summary

In this dissertation, we explore issues of knowledge reuse and support from the perspective of business knowledge-intensive task. Thus, several methods with associated empirical experiments are presented. For achieving task-based knowledge support, we proposed *task-based knowledge support* model to acquire, organize, and disseminate task-relevant information to fulfill the information needs of knowledge workers.

A *task-relevance assessment* approach is proposed to identify workers' information needs on task. The mechanism employs a fuzzy linguistic approach to conduct relevance assessment by the collaboration of knowledge workers. A two-phase assessment is proposed to reduce the burden of assessment load to enable a more effective assessment. Moreover, methods of the adaptation of profiles to track workers' dynamic information needs are proposed in this work. Thus, an *adaptive task-based profiling approach* and a *fuzzy analytical method* are proposed to track workers' dynamic task-needs and identify workers' task-based peer-groups according to the changes of profiles. Accordingly, knowledge workers can obtain task-relevant knowledge with the aid of task-based profiles and peer-groups.

Furthermore, according to our empirical investigation, the knowledge worker engaged in knowledge intensive task (e.g., research projects in academic organizations, project management in firms, etc.) has different information needs during the long-term task performance. Thus, for resolving long-term knowledge support problem, we seek to extend and refine our task-based knowledge support model to fit the problem domain. A *knowledge support model based on task-stage* is proposed. The model provides knowledge support by identifying a worker's information needs at each stage during task performance. Accordingly, techniques for discovering a worker's task-needs, i.e., the task-stage and the task-needs topics at each stage are presented in the dissertation.

Finally, a series of experiments has been conducted to evaluate the proposed model. Furthermore, a $\mathcal{K}$-Support project is carried out in a research institute to evaluate the proposed model. A collaborative task-based $\mathcal{K}$-Support portal is deployed for acquiring, organizing, and disseminating the organization's knowledge resources effectively.

9.2 Future works

This work focuses on providing knowledge support for knowledge-intensive tasks such as thesis works, research projects, project management, and product development. Issues along with the research direction will be addressed.

Context awareness knowledge support: Although we provide a task-view to achieve knowledge support, our current work does not consider the process-aspect and context awareness, as discussed in [1][2][22][24][44]. The process knowledge supports the operations of workflow management systems to manage business processes. The context-based knowledge support utilizes the context of activities, roles, work-related skills, and so on to provide context-aware knowledge access and retrieval. Future studies could extend the proposed approach to support context-aware or process-aware delivery of task-relevant knowledge. Moreover, the information needs of knowledge workers are associated with their roles in undertaken tasks; however, this work does not consider the role/job perspective [5][72] to acquire and disseminate task-relevant knowledge. Future studies could extend the proposed profiling approach by considering role/task to acquire and reuse corporate memory effectively.

Refinement of task domain ontology: In this project, we refer the domain ontology to a classification structure of tasks stored in the information repository [33] [52][56]. Specifically, the domain ontology (DO) is *a simple topic taxonomy that is structured into four levels, including categories, fields, tasks and knowledge items*. In the future, we shall extend our domain ontology to. In the area of knowledge management, the domain ontology also can be expressed as structured link networks are frequently used to represent the organization's knowledge [74]. Besides the topic taxonomy organized in our problem domain, the ontology underlying the $\mathcal{K}$-Support knowledge support portal could be extended to represent the organization's domain-specific knowledge to conduct knowledge support by the utilization of task associated context. The ontological structure can be further extended to specify the knowledge concepts, properties of each concept and semantic relationship between concepts in organizations.

Mining and recommendation techniques in supporting task-relevant knowledge: Due to the limitation of this work, the task-related experts of each task are predefined, as addressed in Section 5. In addition, the relative importance of experts is given the same weight to aggregate the relevance ratings. In the future, we shall consider revising our group decision method with the aid of recommendation and mining techniques in Recommender system. In fact, our lab have investigated the document recommendation in organization with personal folders [36]. That is, we adopt recommendation techniques to provide knowledge workers needed textual documents from other workers folders. Thus, various recommendation methods have been evaluated to analyze the tradeoff between methods. Accordingly, we will employ methods, e.g., collaborative filtering algorithm, demographic profiles of workers, etc, to determine task-related experts and resolve the cold-start problem in system. Thus, the new-system cold-start problem may be resolved by the demographic profiles of workers and the new-user cold-start problem my be resolved by the hybrid recommendation technique.

Moreover, in our on-going work [48], we proposed a task-stage mining method for discovering task-stage needs from historical task sets. Thus, the valuable pieces of knowledge items can be extracted from the mining result. In the future, we will maintain the task-relevant knowledge as the meta knowledge to extend the system capability of finding more task-relevant knowledge by utilizating the context of business historical task. Meanwhile, we will also integrate the proposed task-stage mining technique with our task-stage identification model to investigate the contribution of the task-stage knowledge support model empirically [87].

Computer supported collaborative work: Furthermore, this work focuses on generating task profiles by the collaboration of knowledge workers to analyze the relevance of tasks and codified knowledge. Our work is further enhanced to develop a knowledge support (*K-Support*) system which can stimulate knowledge sharing among task-based peer-groups. Although the *K-Support* system can provide collaborations among knowledge workers through collaborative assessment and knowledge sharing, more computer supported collaborative work (CSCW) is required for successful accomplishment of tasks, especially for complex and volatile tasks. In CSCW environments, groupware is often employed to support collaboration, coordination and communication among groups of people. Notably, this work concentrates on providing task-relevant knowledge without exploring CSCW issues.

Future work will integrate our proposed work with CSCW technology to provide more effective supports for collaborations among knowledge workers. Moreover, some tasks may span across different organizations so that inter-organizational collaboration between knowledge workers is required. This work does not consider inter-organizational collaborations. Further issues regarding this aspect need to be investigated, such as reusing and exchanging task-relevant knowledge across organizations.

Application domain: The proposed task-based knowledge support model may be tailored to other application domain in supporting the execution of long-term knowledge-intensive task-execution. For example, the R&D related work such as project management, intellectual property management, academic researches and industry analysis. We will also seek the other possible application domain to apply the proposed model such as the industry analysis in the project management institution, product development in R&D department and so on.

References

[1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek, "Context-Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project," *International Journal on Information Systems Frontiers (ISF)*, 2(3/4), pp.139-162, 2000a.

[2] A. Abecker, A. Bernardi, H. Maus, M. Sintek, and C. Wenzel, "Information Supply for Business Processes: Coupling Workflow with Document Analysis and Information Retrieval," *Knowledge Based Systems*, 13(1), pp.271-284, 2000b.

[3] A. Agostini, S. Albolino, G. De Michelis, F. De Paoli, and R. Dondi, "Stimulating Knowledge Discovery and Sharing," *Proceedings of the International ACM Conference on Supporting Group Work*, Sanibel Island, Florida, USA, November, pp.248-257, 2003.

[4] H. J. Ahn, H. J. Lee, K. Cho and S.J. Park, "Utilizing Knowledge Context in Virtual Collaborative Work," *Decision Support Systems*, 39(4), pp. 563-583, 2005.

[5] M. Alvarado, M. Romero-Salcedo, and L. Sheremetov, "A Corporative Memory based on the User Profiles," *Expert Systems with Applications*, *26(1)*, pp.87-94, 2004.

[6] J. Andrade, J. Ares, R. Garcia, S. Rodriguez, and S. Suárez, "Lessons Learned for the Knowledge Management Systems Development," *Proceedings of the 2003 IEEE International Conference on Information Reuse and Integration,* pp.471-477, 2003.

[7] R. Baeza-Yates, and B. Ribeiro-Neto, *Modern Information Retrieval*, New York: The ACM Press, 1999.

[8] M. Balabanovi'c, "An Adaptive Web Page Recommendation Service," *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, pp. 378-385, February, 1997.

[9] N. J. Belkin, and W.B. Croft, "Information Filtering and Information Retrieval: Two Sides of the Same Coin?" *Communications of the ACM*, 35(12), pp. 29-38, 1992.

[10] D. Billsus, and M. Pazzani, "User Modeling for Adaptive News Access," *User Modeling and User-Adapted Interaction*, 10(2/3), pp.147-180, 1997.

[11] J.M. Bloodgood, W.D. Salisbury, "Understanding the Influence of Organizational Change Strategies on Information Technology and Knowledge Management Strategies," *Decision Support Systems*, 31(1), pp.55-69, May, 2001.

[12] N. Bolloju, M. Khalifa, and E. Turban, "Integrating Knowledge Management into Enterprise Environments for the Next Generation Decision Support," *Decision Support Systems* 33(22), pp.163-176, June, 2002.

[13] J.S. Brown, and P. Duguid, P. "Organization Learning and Communities of Practice," *Organization Science*, 2(1), pp.40-57, 1991.

[14] K. Byström and K. Järvelin, “Task Complexity Affects Information Seeking and Use,” *Information Processing and Management*, 31(2), pp.191-213, 1995.

[15] A. Celentano, M.G. Fugini, and S. Pozzi, “Knowledge-based Document Retrieval in Office Environment: The Kabiria System,” *ACM Transactions on Information Systems*, 13(3), pp.237-268, 1995.

[16] S.M. Chen, and Y.J. Horng, “Fuzzy Query Processing for Document Retrieval based on Extended Fuzzy Concept Networks,” *IEEE Transaction on System, Man, and Cabernetics* – Part B:Cybernetics, 29(1), pp.96-104, 1999.

[17] T.H. Davenport, and L. Prusak, *Working Knowledge: How Organizations Manage what they know*, Boston MA: Harvard Business School Press, 1998.

[18] N.J. Davies, A. Duke, and A. Stonkus, “OntoShare: Evolving Ontologies in a Knowledge Sharing System, In Davies, NJ, Fensel, D. van Harmelen, F. (Eds.): *Towards the semantic Web – Ontology-based Knowledge Management.* Wiley, London UK, pp.161-176, 2003.

[19] P. De Bra, G.J. Houben, and F. Dignum, “Task-Based Information Filtering: Providing Information that is Right for the Job,” Proceedings of INFWET97, http://www.is.win.tue.nl/~houben/respub/infwet97.html), 1997.

[20] D. Dubis, and H. Prade, H., “Operations on Fuzzy Numbers,” *International Journal of Systems Science*, 9(3), pp.613-626, 1978.

[21] M.J. Eppler, P.M. Seifried, and A. Ropnack, “Improving Knowledge Intensive Process through an Enterprise Knowledge Medium,” *Proceedings of 1999 Computer Personnel Research (ACM SIGCPR)*, New Orleans, LA, USA, pp. 222-230, 1999.

[22] Kurt D. Fenstermacher, *Putting Context in Knowledge Management.* Thesis (Ph.D.), The University of Chicago, 1999a.

[23] Kurt D. Fenstermacher, C. Marlow, “Supporting Consultants with Task-specific Information Retrieval,” *Proceedings of The American Association of Artificial Intelligence*, Orlando Florida, AAAI Press, 1999b.

[24] Kurt D. Fenstermacher, “Process-Aware Knowledge Retrieval,” *Proceedings of the 35th Hawaii International Conference on System Sciences*, Big Island, Hawaii, USA, pp.209-217, 2002.

[25] D. Fensel, S. Staab, R. Studer, F. van Harmelen, and N.J. Davies, “A Future Perspective: Exploiting Peer-to-Peer and the Semantic Web for Knowledge Management,” In Davies, N.J., Fensel, D. van Harmelen, F. (Eds.): *Towards the semantic Web – Ontology-based knowledge management.* Wiley, London UK, pp.245-265, 2003.

[26] G. Fischer, and J. Ostwald, “Knowledge Management: Problems, Promises, Realities, and Challenges,” *IEEE Intelligent Systems*, 16(1), pp.60-73, 2001.

[27] Gartner Group, Knowledge Management Report, 1999.

[28] P.H. Gray, “Problem-solving Perspective on Knowledge Management Practices,” *Decision Support Systems*, 31(1), pp.87-102, 2001a.

[29] P.H. Gray "The Impact of Knowledge Repositories on Power and Control in the Workplace," *Information Technology & People*, 14(4), pp.368-384, 2001b.

[30] J. Hahn, and M. Subramani, "A Framework of Knowledge Management Systems: Issues and Challenges for Theory and Practice," *Proceedings of the 21st International Conference on Information Systems*, Brisbane, Australia, pp. 302-312, 2000.

[31] J.L. Herlocker, and J.A. Konstan, (2001). Content-independent, task-focused recommendation. *IEEE Internet Computing*, 5(6), pp.40-47, 2001.

[32] D. Harman, "Relevance Feedback Revisited," *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Copenhagen, Denmark, pp.1-10, 1992.

[33] N. Hernandez, J. Mothe, and S. Poulain, "Customizing information access according to domain and task knowledge : the OntoExplo System," *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, pp.607-608, 2005.

[34] E. Herrera-Viedma, "Modeling the Retrieval of an Information Retrieval System using an Ordinal Fuzzy Linguistic Approach," *Journal of the American Society of Information Science and Technology*, 52(6), pp.460-475, April, 2001

[35] K. Hoashi, K. Matsumoto, N. Inoue, and K. Hashimoto, "Document Filtering Method Using Non-Relevant Information Profile," *Proceedings of the 23 rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, pp.176–183, 2000.

[36] C.-W. Huang, C.-H. Lai, D.-R. Liu, "Document Recommendation in Organizations with Personal Folders," *Proc. of the Fifth Intl. Conf. on Electronic Business*, HongKong, December 2005.

[37] E. Ide, "New Experiments in Relevance Feedback," in: G. Salton (Eds), *The SMART retrieval system: Experiments in automatic document processing*, Englewood Cliffs, NJ: Prentice Hall, pp.337-354, 1971.

[38] J.S. Jang (Roger), C.T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Upper Saddle River, NJ: Prentice-Hall, 1997.

[39] A. Kankanhalli, F. Tanudidjaja, J. Sutanto, and C.Y. Tan (Bernard), "The Role of IT in Successful Knowledge Management Initiatives," *Communications of ACM*, 46(9), pp.69-73, 2003.

[40] G.J. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, NJ: Prentice Hall Press, 1995

[41] J. Koh, and Kim, Y.G., "Knowledge Sharing in Virtual Communities: An e-Business Perspective," *Expert Systems with Applications*, 26, pp.155-166, 2004.

[42] M. Kracker, "A Fuzzy Concept Network and its Application," *IEEE International Conference on Fuzzy Systems*, San Diego, USA, pp.761-768, 1992.

[43] C. Kuhlthau, *Seeking Meaning*, Noorwood, NJ:Ablex, 1993.

[44] M.M. Kwan, and P. Balasubramanian, "KnowledgeScope: Managing Knowledge in Context," *Decision Support Systems*, 35, 467-486, 2003.

[45] E.S. Lee, R.L. Li, "Comparison of Fuzzy Number based on the Probability Measure of Fuzzy Events," *Computer and Mathematics with Applications*, 15, pp.887-896, 1988.

[46] J. Liebowitz, *The Knowledge Management Handbook*, Boca Raton, FL: CRC Press, 1999.

[47] D.-R. Liu, I.-C. Wu, K.-S. Yang, "Task-based $\mathcal{K}$-Support System: Disseminating and Sharing Task-relevant Knowledge", *Expert Systems with Applications*, 29(2), pp.408-423, 2005a.

[48] D.-R. Liu, I.-C. Wu, and W.-H. Chen, "Reusing Historical Task-relevant Knowledge by Stage-Needs Mining Method," will be submitted to DEXA 2006.

[49] M.L. Markus, "Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situation and Factors in Reuse Success," *Journal of Management Information Systems*, 18(1), pp.57-94, 2001.

[50] T.E. Marshall, and T.A. Byrd, "Perceived Task Complexity as a Criterion for Information Support," *Information & Management*, 34, pp.251–263, 1998.

[51] D.W. McDonald, and M.S. Ackerman, "Expertise Recommender: A Flexible Recommendation System and Architecture," *Proceeding of the ACM 2000 Conference on Computer Supported Cooperative Work,* Philadelphia, PA, pp.231-240, 2000.

[52] S.E. Middleton, N.R. Shadbolt, and D.C. De Roure, "Ontological user profiling in recommender system," *ACM Transactions on Information Systems*, *22*(1), pp. 54-88, 2003.

[53] J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal, "A Multi-level Approach to Intelligent Information Filtering: Model, System and Evaluation", *ACM Transactions on Information Systems*, 15(4), pp. 368-399, 1997.

[54] S. Mukhopadhyay, J. Mostafa., and M. Palakal, "An Adaptive Multi-level Information Filtering System," *In Proceedings of the Fifth International Conference on User Modeling*, pp. 21–28, 1996.

[55] I. Nonaka, "A Dynamic Theory of Organizational Knowledge Creation," *Organization Science*, 5(1), pp.14-37, 1994.

[56] D.E. O'Leary "*Enterprise Knowledge Management*," IEEE Computer, 31(3), pp.54-61, 1988

[57] J. Park, and S. Hunting, *XML Topic Maps: Creating and using Topic Maps for the Web*, Boston MA: Addison-Wesley, 2003.

[58] M. Pazzani, and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," *Machine Learning, 27*, pp.313-331, 1997

[59] W. Pedrycz, “Why Triangular Membership Functions?” *Fuzzy Sets and Systems*, 64(1), pp.21-30, 1994.

[60] M.F. Porter, “An Algorithm for Suffix Stripping,” Program 14(3), pp.130-137, 1980.

[61] C. J. v. Rijsbergen, “Information Retrieval,” Butterworths, London, 1979.

[62] E. Riloff and W. Lehnert, “Information Extraction as a Basis for High Precision Text Classification”, *ACM Transaction on Information System*, 12(3), pp.296-333, 1994.

[63] J.J. Rocchio, *Document Retrieval Systems–Optimization and Evaluation*, PhD thesis, Harvard University, 1966.

[64] J.J. Rocchio, “Relevance Feedback in Information Retrieval,” in: G. Salton (Eds), *The SMART Retrieval System: Experiments in Automatic Document Processing*, Englewood Cliffs, NJ: Prentice Hall, pp.313-323, 1971.

[65] G. Salton, and M.E. Lesk, “Computer Evaluation of Indexing and Text Processing,” in: G. Salton (Eds), *The SMART Retrieval System: Experiments in Automatic Document Processing*, PrenticeHall, Englewood Cliffs, NJ, pp.143-180, 1971.

[66] G. Salton, and M.J. McGill, *Introduction to Modern Information Retrieval*, McGrawHill Book Co., New York, 1983.

[67] G. Salton, and C. Buckley, “Term Weighting Approaches in Automatic Text Retrieval,” *Information Processing & Management*, 24(5), pp.513-523, 1988.

[68] G. Salton, and C. Buckley, “Improving Retrieval Performance by Relevance Feedback,” *Journal of the American Society for Information Science*, 41(4), pp.288-297, 1990.

[69] T. Saracevic, “Relevance: A Review of and a Framework for Thinking on the Notion in Information Science,” *Journal of American Society for Information Science*, 26, pp.321-343, 1975.

[70] F. Sebastiani, “Machine Learning in Automated Text Categorization: A Survey,” *Technical report*, Istituto di Elaborazione dell'Informazione, C.N.R., Pisa, Italy, 1999.

[71] B. Shapira, P. Shoval and U. Hanani, (1999), "Experimentation with an Information Filtering System that Combines Cognitive and Sociological Filtering Integrated with User Stereotypes," *Decision Support Systems*, 27, pp.5–24, 1999.

[72] M. Shen, and D.-R. Liu, “Discovering Role-relevant Process-views for Disseminating Process Knowledge,” *Expert Systems with Applications, 26(3)*, pp.301-310, 2004.

[73] M. Spies, A.J. Clayton, and M. Noormohammadian, “Knowledge Management in a Decentralized Global Financial Services Provider: a Case Study with Allianz Group,” *Knowledge Management Research & Practice* 3, pp.24–36, 2005.

[74] K. Steiner, W. Essmayr, and R. Wagner, “Topic Maps - An Enabling Technology for Knowledge Management,” *Proceedings of 12th International Workshop on Database andExpert Systems Applications (DEXA'01)*, pp. 472-476, September 3-7 2001.

[75] R.S. Taylor, “Question-Negation and Information Seeking in Libraries,” *College and*

Research Libraries, 29, pp.178-194, 1968.

[76] P. Vakkari, "Relevance and Contribution Information Types of Searched Documents in Task Performance," *Proceedings of the 23rd Annual International ACM SIGIR Conference*, pp.2-9, 2000.

[77] P. Vakkari, M. Pennanen, and S. Serola, "Changes of Search Terms and Tactics While Writing a Research Proposal: A Longitudinal Case Study ", *Information Processing and Management*, 39(3), pp.445-463, 2003.

[78] P.L. Wang, and D. Soergel, "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), pp.115-133, 1998.

[79] P.L. Wang, and M.D. White, "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), pp.98-114, 1999.

[80] H. Widyantoro, T. R. Ioerger, and J. Yen, "Learning User Interest Dynamics with a Three-Descriptor Representation", *Journal of the American Society for Information Science*, 52(3), pp.212-225, 2001.

[81] C. Wei, P. Hu, and H. H. Chen, "Design and Evaluation of A Knowledge Management System", *IEEE Software*, 19(3), pp.56-59, 2002.

[82] Wiig, (1993), *Knowledge Management Foundation.* Schema Press.

[83] I.H. Witten, A. Moffat, and T.C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*, 2nd edn., Morgan Kaufmann Publishers, Los Alto, USA, 1999.

[84] M. Wolverton, "Task-based Information Filtering," *ACM Computing Surveys*, 31(2), Article No. 10, 1999.

[85] I.-C. Wu, and D.-R. Liu, "A Collaborative Relevance Feedback Approach to Task-driven Recommendation," *Proc. of the Seventh Pacific-Asia Conf. on Information Systems*, Adelaide, Australia, pp. 52-67, 2003.

[86] I.-C. Wu, K.-S. Yang, W.-H. Chen, and D.-R. Liu "Deployment of Task-based Knowledge Support System," *Proc. of the 10th Conference on Information Management Research and Practice (CSIM2004)*, Taiwan, Dec. 2004.

[87] I,-C. Wu, Duen-Ren Liu, and Pei-Chen Chang, "Proactive Codification-based Knowledge Support by Exploitation of Task Stage Identification," under preparation for journal publication, 2005.

[88] Y. Ye, and G. Fischer, "Supporting Reuse by Delivering Task-Relevant and Personalized Information," *Proceedings 24th International Conference on Software Engineering*, Orlando, Florida, USA, pp. 513-523, 2002.

[89] M.H. Zack, "Managing Codified Knowledge," *Sloan Management Review*, 40(4), pp.45-58, 1999.

[90] L.A. Zadeh, "Fuzzy Sets," *Information and Control*, 8(3), pp.338-353, 1965.

[91] L.A. Zadeh, (1975). "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning," Parts 1, 2, and 3. *Information Sciences,* 8(2), pp.199-249; *8*(3), pp.301-357; *9*(1), pp.43-80, 1975.

Appendix A. Basic Concepts

A.1 Fuzzy Linguistic and Fuzzy Number

Definition I [91]**:** A **linguistic variable** is expressed as a quintuple (*S*, *E(S)*, *U*, *G*, *M*) where *S* denotes the *name* of the variable; *E(S)* is the *linguistic term* of *S*, namely the set of its linguistic values range over a universe of discourse *U*; *G* is a *syntactic rule* (a grammar) which generates the linguistic term set in *E(S*); and M is a *semantic rule* that assigns meaning, *m(e*), to each linguistic term *e* in *E* with a fuzzy set on *U*.

From *Definition I*, a linguistic variable, *Relevance*, is defined to represent the degree of relevance between items (tasks or categories) assessed by evaluators. *E*(*Relevance*) is characterized using a fuzzy set of a universe of discourse *U*=[0,1], in which six linguistic terms řj and their associative semantic meanings m(řj) are defined as follows:

E(*Relevance*) = { ř0 = *Very Low (VL)*, ř1 = *Low (L)*, ř2 = *Normal (N)*, ř3 = *High (H)*, ř4 = *Very High (VH)*, ř5 = *Perfect (P)*}

where m(ři) < m(řj), for i < j, and all m(řj) are distributed in [0,1].

The fuzzy linguistic approach models the meaning of each term using fuzzy numbers, as defined in *Definition II[20]*. The fuzzy number plays a fundamental role in formulating the semantic meaning of the linguistic term, which represents an approximate value of the linguistic variable.

Definition II [20]**:** A **fuzzy number** $^{\alpha}\widetilde{Z}$ is a "normal" and "convex" fuzzy set defined on the set $\mathbb{R}$ and $\widetilde{Z}$ is a closed interval for every $\alpha \in (0,1]$. The membership function $f_{\widetilde{Z}}(x)$ of the triangular fuzzy number (TFN), $\widetilde{Z} = (l,\ m,\ r)$, is presented in Eq. 3 [59].

$$f_{\tilde{Z}}(x) = \begin{cases} (x-l)/(m-l) & l \le x \le m \\ (r-x)/(r-m) & m \le x \le r \\ 0 & otherwise \end{cases} \quad \textbf{(A.1)}$$

This work adopts the center of area (COA) method to calculate fuzzy numbers, owing to its simplicity and practicability. The COA method calculates the fuzzy mean under uniform probability distribution assumption (Lee & Li., 1988)[45]. If the fuzzy number *Ũ* is triangular, where *Ũ*=(*l*,*m*,*r*). The crisp rating can be derived by the equation: $CV(\tilde{U})=[(r-l)+(m-l)]/3+l$.

A.2. Fuzzy Relations

Definition III [16][40]**:** Given an n-by-n fuzzy similarity relationship matrix $\boldsymbol{S}$ which represents the fuzzy relation among $\boldsymbol{U}$, a set of workers, where $|\boldsymbol{U}| = n$. A transitive max-min closure $\boldsymbol{S_T}$ of the similarity matrix $\boldsymbol{S}$ is derived as $\boldsymbol{S_T} = \boldsymbol{S}^y$ by a sequence of max-min operations on the relation matrix until $\boldsymbol{S}^{\boldsymbol{y}}=\boldsymbol{S}^{\boldsymbol{y+1}}=\ldots=\boldsymbol{S}^{\infty}$. Notably, $\boldsymbol{S}^y=\boldsymbol{S}^{y-1}\circ\boldsymbol{S}^{y-1}$, where y is an integer, $1\leq y \leq n$-1 and ° denotes a fuzzy max-min operation. The max-min composition and max operator for set unions are used to derive the transitive max-min closure $\boldsymbol{S_T}$. The fuzzy max-min operation is defined as shown in Eq. A.2.

$$\tilde{\varsigma}^{y}(E_i, E_j) = \max_{E_u \in U} \min(\tilde{\varsigma}^{y-1}(E_i, E_u), \tilde{\varsigma}^{y-1}(E_u, E_j)) \qquad \textbf{(A.2)}$$

where $\tilde{\varsigma}^{y}(E_i, E_j)$ represents an element in $\boldsymbol{S}^y$ and $\tilde{\varsigma}^{y-1}(E_i, E_u) / \tilde{\varsigma}^{y-1}(E_u, E_j)$ represents an element in $\boldsymbol{S}^{y-1}$.

Appendix B. Details of System Evaluation

Herein, we listed the details of system evaluation in Section 6.6.2. And we summarize the observations form the given data. As we have addressed in Section 6.5, we choose 12 evaluators to evaluate the developed system based on the proposed methods. There are 7 experienced workers and 5 novices to participate the evaluation task.

Table 22 shows the average new items (tasks and documents) viewed by the evaluators. In addition, the average new items (documents) viewed and relevant is also given in the table 22. The result revels that novice seems more satisfied with the support document because the percentage of relevant items to the viewed items of novices higher than that of experienced workers. If we took a further look, we may discover that experienced workers tend to give more irrelevant rating to the viewed items. In addition, the novices also tend to give more "Normal" ratings than that of experienced workers. Table 23 shows the average number of "irrelevant" and "normal" ratings on viewed documents for two user groups. That's the reason that we infer the experienced workers seems more knowledgeable on the executing task and use more filtering strategy than that of novices.

Table 22. Viewed / relevant of supporting items

Conditions		Experienced workers		Novices	
		Task	Document	Task	Document
Adapted K-Delivery	Average Number of newt items viewed	1.571	9.571	1.600	6.600
	Average number of new items viewed and relevant	1. 000 (63.65%)	8.714 (85.40%)	1. 600 **(100%)**	5.800 **(87.88%)**
K-Sharing	Average Number of newt items viewed	5.571	5.429	4.400	8.000
	Average number of new items viewed and relevant	2. 571 (46.15%)	4.286 (78.95%)	3. 000 **(90.9%)**	7.800 **(97.5%)**

Table 23. Irrelevant/ Normal Ratings of viewed items

Conditions		Experienced workers		Novices	
		Task	Document	Task	Documen
Adapted K-Delivery	Average Number of Irrelevant Rating	**0.714**	**0.857**	0.200	0.400
	Average Number of Rating is "Normal"	0.143	1.286	**0.600**	**1.800**
K-Sharing	Average Number of Irrelevant Rating	**2.000**	**1.286**	0.600	0.200
	Average Number of Rating is "Normal"	2.714	1.429	**1.800**	**2.200**