知識流探勘與文件推薦之整合研究(2/3)
Research on Knowledge Flow Mining and Document Recommendation (2/3)

計劃編號：NSC 96-2416-H-009-007-MY3
執行期間：97 年 8 月 1 日至 98 年 7 月 31 日
主持人：　劉敦仁　　交通大學 資訊管理研究所
計劃參與人員：賴錦慧、陳韋孝、陳宇軒、林子瑋、邱璇

摘要

知識是建立組織核心競爭力的一個重要資源，透過知識流，使得組織知識在組織內傳播和累積，以支援工作者的工作。執行工作的過程中，知識工作者會參與許多以工作為基礎之群組，並與其合作以滿足工作需求。因此本研究針對以工作為基礎之群體，整合資訊檢索與資料探勘等技術，提出以群組為基礎之知識流探勘演算法。透過演算法進行群組知識流的探勘與建構，並建立知識圖來表示群組知識流，以發掘群組中具有相似工作需求之知識工作者的知識參考行為。計畫的執行進度包括：(1)分析知識工作者之知識流，建立以工作為基礎之知識群組。(2)分析群體中工作者的知識流，提出群體知識流探勘的演算法，並使用知識圖來表示群體知識流。(3)實作群體知識流探勘的雛型系統。

關鍵詞：知識流探勘、知識分享、文件推薦、協同式過濾、序列規則探勘

Abstract

Knowledge is the most important resource to create core competitive advantages for an organization. Such knowledge is circulated and accumulated by a knowledge flow (KF) in an organization to support worker's tasks. Workers may cooperate and participate in several task-based groups to fulfill their needs. In this project, we propose a group-based knowledge flow mining algorithm which integrates information retrieval and data mining techniques for mining and constructing the group-based KF (GKF) for task-based groups. The GKF is expressed as a directed knowledge graph to represent the knowledge referencing behavior for a group of workers with similar task needs. The research progress of this project includes the following. (1) Analyzing the knowledge flows of workers, and building task-based knowledge groups; (2) Analyzing workers' knowledge flows in a group, proposing the algorithm for mining the GKF and represent the GKF as a knowledge graph; (3) Implementing a prototype system for mining the group-based knowledge flow.

Keywords: Knowledge Management, Knowledge Flow, Process Mining, Data Mining

1. Background and research objective

Knowledge and expertise in an organization are generally codified in textual documents, e.g., papers, manuals and reports, and preserved in the knowledge database. A large number of such codified knowledge is circulated and accumulated in an organization to support knowledge workers engaged in tasks and activities. To preserve, share and reuse such a valuable asset, organizations need to adopt appropriate strategies to support knowledge workers intelligently [9]. In task-based business environments, the working knowledge may flow among workers in an organization, while the process knowledge may flow among various tasks [15]. Knowledge flow reflects the level of knowledge cooperation between workers or processes, and influences the effectiveness of teamwork/workflow. Furthermore, knowledge flow (KF) can be used to represent the information needs of knowledge workers and the knowledge flow-based recommendation method can proactively deliver the task-relevant topics and documents to workers [7].

To work efficiently, workers with similar expertise and experiences may join the same group as a task-based group and collaborate with each other to fulfill a task. These workers have similar referencing behavior in referencing and learning knowledge and share task knowledge to fulfill their information needs. Modeling the referencing behavior for a group of workers is an important issue to facilitate knowledge sharing. However, to our knowledge, there is no appropriate approach to analyze and construct the KF according to the perspective on a group's information needs.

A group-based knowledge flow (GKF) represents the information needs and the common referencing behavior for a group of workers. Workers in the same group have similar information needs and can share their task knowledge with each other to effectively fulfill the task. We propose an algorithm which integrates information retrieval and data mining techniques for mining and constructing the KFs for groups. In our previous work, we have proposed an approach of KF mining [7] to identify the KF for each knowledge worker. This work further extends our previous work and focuses on how to discover a group-based knowledge flow from workers' KFs. We present a GKF mining algorithm to discover the frequent referencing behavior for a group of workers. The concept of graph theory is used to visualize the GKF as a knowledge graph, where a vertex and an edge indicate a topic and a direct flow relation of knowledge respectively. Finally, we implement a prototype of knowledge flow mining system to demonstrate our proposed method.

2. Research results

This work proposes an algorithm for mining the group-based knowledge flow (GKF) from workers with similar KFs. The discovered GKF model is used to represent the information needs and knowledge referencing behavior for a group of workers. According to GKF, representative frequent referencing paths can be suggested as common knowledge referencing patterns for the group. Such patterns can be provided as learning references to help novice workers conduct tasks. Finally, we implement a prototype system to demonstrate our proposed algorithm. Our method for mining the group knowledge flow can enhance the organizational learning and facilitate knowledge management, sharing, and reuse in the context of collaboration and teamwork.

3. Related work

3.1. Knowledge flow

The concept of knowledge flow has been applied in various domains, e.g., scientific research, communities of practice, teamwork, industry, and organizations [3, 14]. Scientific articles represent the major medium for disseminating knowledge among scientists to inspire new ideas [14]. KF in weblogs is regarded as a communication pattern whereby the post of one blogger links to the post of another blogger to exchange knowledge [3]. Moreover, knowledge flow in communities of practice helps members share their knowledge and experience about a specific domain to complete their tasks [10].

3.2. Information retrieval and task-based knowledge support

Information retrieval (IR) deals with the representation, organization, and storage of knowledge, and facilitates access to, information items [4]. Essentially, IR focuses on searching for and indexing a large number of documents, and presenting users with data that meets their information needs. In the IR domain, the vector space model [11] is typically used to represent documents as vectors of index terms, where the weights of the terms are measured by the *tf-idf* approach. The *tf-idf* weight is then used to estimate the importance of a term in a document stored in the corpus; *tf* denotes the occurrence frequency of a particular term in the document, while *idf* denotes the inverse document frequency of the term.

Information filtering with a similarity-based approach is often used to locate knowledge items relevant to the task-at-hand. The discriminating terms in a task are usually extracted from a knowledge item/task to form a task profile which is used to model a worker's information need [6, 8].

3.3. Process mining

In workflow system, the process mining technique can be used to extract a structural process description from a set of real process execution [13], infer the relations between tasks/activities and automatically generate a process model from event-based data (log) [2, 12]. The relations between processes (tasks or activities) are defined as casual/parallel relations and are modeled by a directed graph [2]. In a directed graph, the edges specify the potential flow of control from one activity to another, while the vertices represent the activities in a process.

4. Mining group-based knowledge flow

4.1. Overview and definitions

According to the access time of knowledge, a knowledge flow (KF) represents the information needs and the accumulated task-related knowledge for a knowledge worker when he/she perform a task. Workers with similar KFs can be regarded as an interest group with similar information needs and task-related knowledge. In such group, workers can share their referencing behavior and work knowledge (documents) during task performance. A group-based knowledge flow (GKF) can be discovered from the KFs of workers to model their common referencing behavior.

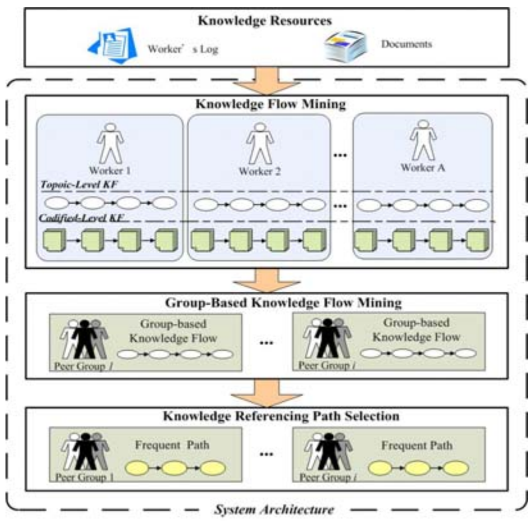

Figure 1. The overview of mining the group-based knowledge flow

Figure 1 shows an overview of our proposed method for mining the GKF. According to the KF of workers, workers with similar KFs are clustered together to form a task-based group. We propose a group-based KF mining algorithm based on the concept of process mining and graph theory. Our algorithm can identify the common information needs and referencing patterns from the KF of a group of workers, and then build a group-based knowledge flow (GKF) represented as a directed acyclic graph. We also proposed a method to identify frequent knowledge paths which represent the frequent referencing patterns of a group. Such frequent knowledge referencing paths can be used to help novice workers learn and utilize task-related knowledge.

A GKF model is defined formally as below.

Definition 1 (KF): Knowledge flow (KF) is defined as an evolution of a worker's information needs, preferences and referencing behavior for obtaining the task-related knowledge. A KF consists of two levels: a codified level and a topic level. Let a worker's personal knowledge flow be $KFlow_w$ = {TKF_w, CKF_w}, where TKF_w is the topic-level KF of worker *w* for a specific task, and CKF_w is the codified-level KF of worker *w* for a specific task. Both the topic-level KF and the codified-level KF are time-ordered sequences based on the access time of the topics and documents. The topic-level KF and the codified-level KF are defined in Definition 2 and Definition 3 respectively.

To fulfill a task, a worker may reference documents (codified knowledge) from various topic domains at different time. During the task execution, such knowledge flows among documents and topics for stimulating a worker to access the next task-related topics and documents. Such KF is identified from a worker' historical work log which records the accessed documents and the access time of them when the worker executes a task.

Definition 2 (Codified-level KF): The codified-level KF indicates that the codified-level knowledge flows among documents based on the referenced time when a worker performs a task. The codified-level KF is defined as $CKF_w = < d_w^{t_1}, d_w^{t_2}, \cdots, d_w^{t_f} >$ where $t_1 < t_2 < \cdots < t_f$ and $d_w^{t_j}$ is the document that worker *w* accessed at time t_j for a specific task. The codified-level KF contains a sequence of documents ordered by the time the documents were accessed.

Definition 3 (Topic-level KF): The topic-level KF represents that the knowledge flows among various topic domains. $TKF_w = < TP_w^{t_1}, TP_w^{t_2}, \cdots, TP_w^{t_f} >$ denotes the topic-level KF of worker *w*, where $TP_w^{t_j}$ is the corresponding topic of the document that worker *w* accessed at time t_j for a specific task. The topic-level KF is generated from the codified-level KF by mapping the codified knowledge into the corresponding topic domain and is represented as a sequence of topics which is arranged by the access time.

Definition 4 (Direct Flow Relation): In a pattern of the topic-level KF, topic *x* is followed by topic *y* if the accessed time of topic *x* is earlier than that of topic *y*. A topic *x* is directly followed by topic *y* if *x* is followed by *y* and there does not exist a distinct topic *z* such that *x* is followed by *z* and *z* is followed by *y*. The relation between topics *x* and *y* is a directed flow relation $x \rightarrow y$, if *x* is directly followed by *y*.

To represent the KF for a group, we exploit the concept in the graph theory to model the GKF as a directed knowledge graph.

Definition 5 (Knowledge Graph): A knowledge graph is defined as $G = (V, E)$, where *V* is a finite set of vertices in which a vertex denotes a topic in the knowledge domain. *E* is a finite set of directed edges connecting two topics. Each edge $e_{x,y}$ denotes the knowledge flow from topic *x* to topic *y*.

Definition 6 (Group-based Knowledge Flow; GKF): The GKF is generated from a group of workers who are in the same cluster and have similar KFs. The GKF is composed of three set, including a directed knowledge graph, a set of workers and a set of topic-level KFs and is defined as *GKF* = {*DG*, *W*, *TKF*} where *DG* is a directed knowledge graph; *W* is a set of *n* workers who have similar *KF*; $TKF = \{TKF_j \mid \forall j, j = 1 \cdots n\}$ is a set of topic-level KF of workers in *W*.

Definition 7 (Knowledge Referencing path): Given a directed knowledge graph *DG* of GKF, if there is a path from a starting vertex to an end vertex, such path is a knowledge referencing path. Let p_k = {*s*, *d*, V_{p_k}, E_{p_k}} be a referencing path in the graph of GKF, where *s* is a start vertex, *d* is an end vertex, V_{p_k} is a set of topics included in the path p_k, and E_{p_k} is a set of edges where each edge e_{v_i,v_j} is an ordered pair (v_i, v_j), where v_i and $v_j \in V_{p_k}$.

4.2. Worker clustering

Workers with similar KFs are clustered together because they have similar task knowledge and referencing behavior. A worker's KF consists of essential knowledge for performing tasks: topic-level knowledge and codified-level knowledge, which are derived from a worker's log and are arranged by the referenced time. To cluster workers, we apply the CLIQUE clustering method [1] to cluster knowledge workers based on their KFs. First, a similarity matrix of workers is built before clustering. The entry of the similarity matrix is the degree of similarity between two workers. The similarity of two compared KFs is evaluated by a hybrid similarity measure [7] which combines the KF alignment similarity and the aggregated profile similarity. A worker may have a variety of information needs or task knowledge. Thus, a worker may participate in different clusters.

4.3. The GKF mining algorithm

To identify the GKF model from a set of KF, we propose an algorithm without considering topic cycles in KFs. This algorithm can identify the *DG* of a GKF from a set of KFs for a group of workers who have similar KFs. We consider *DG* as a directed acyclic graph. Specifically, the algorithm assumes that there is no duplicate topic appearing in a KF and there are no topic cycles in the graph *DG*. For workers in a specific group (task) *k*, their KFs are the input of this algorithm, while the graph of the GKF_k is the output result.

Our algorithm ensures that any vertex *v* in the graph is reachable such that at least one path exists from the start vertex through the vertex *v* to the end vertex. The frequent edges representing frequent knowledge flow between topics are preserved in the GKF. On the contrary, the infrequent edges with weights lower than a user-specified threshold are removed from the graph.

Table 1. The algorithm for mining the *DG* of a GKF without considering topic cycles

```
GKF mining Algorithm
Input: A set of n workers W and their KFs, TKF = {TKF_w|
       w=1…n};
Output: GKF_k ={DG, W, TKF };
Let DG={V, E} be a directed knowledge graph, where
V=ϕ , E=ϕ ;
Add a start vertex s to V;
Add an end vertex d to V;
for each TKF_w in TKF {
  for each topic v_y in TKF_w according to the sequence order)
      {
      if (v_y does not exist in V)    //add a vertex
         Add v_y to V;
      if (v_y is the first vertex in TKF_w)
         Add e_{s,y} to E;
      else if (v_y is the last vertex in TKF_w)
          Add e_{y,d} to E;
      for each vertex v_x ∈ V and v_x →v_y in TKF_w
          if (e_{x,y} ∉ E) Add e_{x,y} to E;
   }
}
Calculate the weights for all edges in E;
L = Topological Sorting (V, E);
Edge Deletion (L, V, E);
```

The details of the proposed algorithm are shown in Table 1. To generate a GKF model for a specific group *k*, a set of KFs which belong to a set of workers is considered as the input of the algorithm. Each topic in the KF is regarded as a vertex in the graph, while each direct flow relation between two topics in TKF is regarded as an edge. For example, given a KF as <A, B, E, C>, the four topics A, B, E and C are four vertices and three flows $e_{A,B}$, $e_{B,E}$, and $e_{E,C}$ are three directed edges in the graph. Note that the edge, e.g., $e_{A,B}$, is used to direct the flow from one topic to the other, e.g. from topic A to topic B.

The algorithm has several steps to build the GKF model. First, we add a start vertex and an end vertex first in the directed acyclic graph. Second, each topic (vertex) in each KF is added into a vertex set *V* if this vertex does not exist in *V*. Then, the edges related to the inserting vertex are added to the edge set *E* if they do not exist in *E*. There are three strategies for generating edges and deciding which edges are inserted into *E*. If the vertex *y* is the first vertex in a KF, the edge $e_{s,y}$ from the starting vertex *s* to the vertex *y* is inserted into *E*; if $x \rightarrow y$ exists in TKF_w, the edge $e_{x,y}$ is added to *E*; and if the vertex *y* is the last topic in the KF, edge $e_{y,d}$ from the vertex *y* to the ending vertex *d* is inserted into *E*. After all vertices and their related edges are added to *V* and *E* respectively, an initial *DG* of the GKF model is built.

4.3.1. Measuring the importance of an edge

Some edges in the graph may be infrequent in a KF and are not appropriate to represent the referencing behavior of workers in the group. To deal with this problem, the algorithm calculates the weight of each edge in *E* to evaluate its importance in a GKF. The weighting function measures the importance of an edge in a GKF model, as defined in Eq. 1.

$$we_{x,y} = \frac{\sum_{TKF_w \in TKF} f_{x\rightarrow y}}{N_{TKF}}, \text{ where } \quad f_{x\rightarrow y} = \begin{cases} 1, if \;\; x \rightarrow y \in TKF_w \\ 0, otherwise \end{cases} \qquad \textbf{(1)}$$

where $we_{x,y}$ is the weight of the edge $e_{x,y}$ which appears in the GKF model and represents a direct flow from the vertices *x* to *y*; TKF_w is the topic-level KF of a worker *w*; $f_{x\rightarrow y}$ is 1 if the direct flow relation $x\rightarrow y$ exists in TKF_w; otherwise, it is 0; and N_{TKF} is the number of the topic-level KF in a group. In a specific TKF_w, the direct flow relation of two topics, e.g. $x\rightarrow y$, can be represented as an edge, e.g. $e_{x,y}$, in a GKF. The more frequent the direct flow relation in a group, the more important the edge in the graph.

According to the weights of edges in GKF, the topological sorting and the edge deletion procedures are designed to remove the infrequent edges from the graph. The topological sorting procedure is used to sort all vertices in topological order, while the edge deletion procedure is used to check all edges based on the sorting result and then remove infrequent edges. In the following, we will elaborate these two procedures in detail.

4.3.2. The procedure of topological sorting

Topological sorting [5] orders the vertices of a directed acyclic graph in a topological order, such that for any two vertices *x* and *y*, if *x* is a predecessor of *y* in the graph then *x* precedes *y* in the topological ordering.

In the edge deletion procedure, the algorithm checks the incoming edges for each vertex in the topological order and determines which edges should be removed. Sorting vertices of a *DG* in topological order ensures that a preceding vertex is processed before a successive vertex. Therefore, when performing the edge deletion, removing an infrequent edge from a vertex will not affect the reaching ability of its preceding vertices.

4.3.3. The procedure of edge deletion

According to the result of topological sort, the edge deletion procedure examines all vertices in topological order and then removes the infrequent edges with weighting values lower than a user-specified threshold, as shown in table 2. The inputs of this procedure are the sorted list *L* and the *DG* of the GKF. All the vertices in *L* are examined in topological order. For each vertex, the algorithm checks the incoming edges of the vertex in ascending order by their weights. The incoming edges with weights lower than user-specified threshold are candidates to be removed.

Table 2. The procedure of edge deletion

```
Edge Deletion (L, V, E)
 Q = ϕ ; // the checked set of vertices
 for each vertex v_y in a topological order L {
     for each incoming edge e_x,y of v_y according to its
         weight in ascending order {
       if (we_x,y < threshold θ) {
          Remove e_x,y from E;
       if (no path p_s,y exists from s to v_y)
          Add e_x,y to E;
       else { //checking if there exists a path from
               // vertex v_j to the end vertex d
         for each v_j in Q {
            if (no path p_j,d exists from v_j to d) {
               Add e_x,y to E; break; }
         }
       }
     }
 }
 Add v_y to Q;
 }
```

To remove an infrequent edge, the procedure has to check if removing such edge will make the vertex unreachable from the start vertex or result in other vertices that can not reach the end vertex. If such situations occur, the removed edge should be added back. For a vertex v_y, if one of its incoming edges is removed and there exists no other path from the start vertex to the vertex v_y, the removed edge should be added back to the edge set *E*. Inaddition, according to the topological order, the checked vertices processed before v_y are examined again to make sure that for each checked vertex v_j in *Q*, there exists a path from the vertex to the end vertex. If removing the edge violates the above condition, such edge should be added back to the edge set *E*.

The procedure of edge deletion can ensure that any vertex in GKF can be traversed from the start vertex. Also, there exists at least one path from each vertex to the end vertex. Moreover, we can obtain a most frequent path in GKF to assist workers in learning group's knowledge.

4.4. Identify knowledge referencing paths

We develop a method for identifying frequent knowledge referencing paths from a GKF, which is the common knowledge referencing behavior of the group. A path score derived from the weights of edges in a path is used to evaluate each knowledge path. Paths with scores higher than a user-specified threshold are regarded as frequent knowledge paths in a GKF. Specifically, such knowledge paths (patterns) represent the frequent knowledge referencing behavior of workers and important knowledge flows in a group. The paths with scores higher than a user-specified threshold are selected for the group.

A path score indicates the degree of importance of a path based on the weights of the edges in a path as defined in Eq. 2.

$$ps_i = Min\{we_{x,y} \mid \forall e_{x,y} \ in \ path_i\} \qquad \textbf{(2)}$$

Where ps_i is the path score of the path *i*, $we_{x,y}$ is the weight of edge $e_{x,y}$ which represents a direct flow relation between vertex *x* and vertex *y* in path *i*. A path score is derived from a minimal weight among edges in the path to indicate the important degree of a path. Note that the edge weight is defined in Eq. 1, which denotes the importance of the direct flow in a GKF. The large edge weight means that the edge is highly significant for the group of workers.

5. The prototype system

Our prototype system for group-based KF mining is implemented by using the software: Microsoft Visual Studio 2005 (with C#) that is used to develop the system and Microsoft SQL Server 2005 that is used as the database system for storing a dataset. We use a dataset from a real application domain, a research laboratory of a research institute. Each worker may fulfill several tasks e.g. conducting research projects and writing research papers, while the research documents are the needed codified knowledge when workers perform tasks. Documents in the dataset are grouped into eight clusters by using the single-link clustering method. Based on the clustering results, topic-level KFs are generated by mapping the codified knowledge into its corresponding clusters. Then, the two levels of KF, the topic-level KF and the codified-level KF, are generated to describe the information needs of a worker.

Our system can be applied not only in a research laboratory, but also in a knowledge intensive organization to help workers learn and obtain knowledge. The system provides an interface to show the topic-level KF for all workers and the result of worker clustering. To simplify the representation of the KF, we use a number to denote a topic domain that consists of topic-related terms.

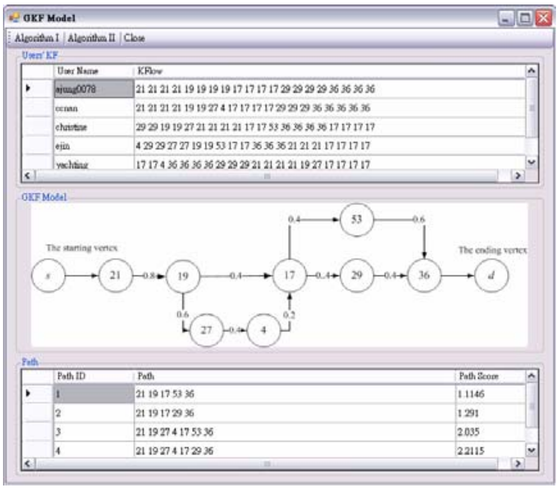

Figure 2. The generalization of the GKF model

Our system uses GKF mining algorithms to build the GKF model for each group, as shown in Figure 2. All workers in a cluster have similar KFs to each other. Then, these workers' KFs are used to build the GKF to characterize the referencing behavior for the group. The GKF is shown in the frame of the GKF model. Each circle is a topic domain represented by a number, while each directed edge is a flow of knowledge between two topics. The number on the arrow is a weight to indicate the importance of an edge. According to the weight of edges in a path, the system can obtain paths with scores larger than a user-specified threshold as frequent knowledge referencing patterns for the group.

References

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proceedings of the international conference on Management of data (ACM SIGMOD)*, 1998, pp. 94-105.

[2] R. Agrawal, D. Gunopulos, F. Leymann, and G. Boblingen, "Mining Process Models from Workflow Logs," in *6th International Conference on Extending Database Technology (EDBT'98)*, Valencia, Spain, 1998, pp. 469-483.

[3] A. Anjewierden, R. de Hoog, R. Brussee, and L. Efimova, "Detecting knowledge flows in Weblogs," in *13th International Conference on Conceptual Structures (ICCS 2005)*, 2005, pp. 1-12.

[4] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston: Addison-Wesley, 1999.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*: MIT Press, 2001.

[6] H. Holz, H. Maus, A. Bernardi, and O. Rostanin, "A Lightweight Approach for Proactive, Task-Specific Iinformation Delivery," *Proceedings of the 5th International Conference on Knowledge Management (I-Know)*, 2005.

[7] C.-H. Lai and D.-R. Liu, "Knowledge Flow-Based Document Recommendation for Knowledge Sharing," in *The Second International Conference on Knowledge Science, Engineering and Management (KSEM07)*, Melbourne, Australia, 2007, pp. 325-335.

[8] D. R. Liu, I. C. Wu, and K. S. Yang, "Task-based K-Support system: disseminating and sharing task-relevant knowledge," *Expert Systems With Applications*, vol. 29, 2, pp. 408-423, 2005.

[9] I. Nonaka and H. Takeuchi, *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*: Oxford University Press, 1995.

[10] O. M. Rodriguez, A. I. Martinez, J. Favela, A. Vizcaino, and M. Piattini, "Understanding and supporting knowledge flows in a community of software developers," in *International Workshop on Groupware (CRIWG)*, 2004, pp. 52-66.

[11] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, 5, pp. 513-523, 1988.

[12] W. M. P. van der Aalst and A. Weijters, "Process mining: a research agenda," *Computers in Industry*, vol. 53, 3, pp. 231-244, 2004.

[13] W. M. P. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: discovering process models from event logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, 9, pp. 1128-1142, 2004.

[14] H. Zhuge, "Discovery of knowledge flow in science," in *Communications of the ACM*. vol. 49: ACM Press New York, NY, USA, 2006, pp. 101-107.

[15] H. Zhuge, "A knowledge flow model for peer-to-peer team knowledge sharing and management," *Expert Systems with Applications*, vol. 23, 1, pp. 23-30, 2002.