Discovering Workflow-Aware Virtual Knowledge Flows for Knowledge Dissemination

Minxin Shen and Duen-Ren Liu

Institute of Information Management, National Chiao-Tung University
1001 Ta Hsueh Road, Hsinchu 300, Taiwan
{shen,dliu}@iim.nctu.edu.tw

Abstract. In order to effectively disseminate task-relevant and process-scope knowledge, knowledge-intensive enterprises adopt knowledge flows to explicitly represent workers' knowledge needs and referencing behavior of codified knowledge during the execution of business tasks. However, due to differences in expertise and experience, individual workers impose varied knowledge needs on the knowledge flows directed by the workflows they participate in. This study proposes a model of *workflow-aware knowledge-flow views*, i.e. virtual knowledge flows abstracted from workflow-driven knowledge flows, to provide adaptable knowledge granularity. Moreover, a text mining approach is developed to derive knowledge-flow views from codified knowledge objects of knowledge flows, such as documents. Both task knowledge semantics and task execution sequences are utilized to evaluate the degrees of workers' knowledge demands in workflow contexts. Knowledge management systems can thus present different abstracted knowledge flows to diverse workflow participants, and facilitate knowledge sharing and collaboration.

Keywords: knowledge flow, workflow, knowledge management, text mining.

1 Introduction

In knowledge-intensive work environments, workers require task-relevant knowledge and documents to support their execution of tasks. Thus, effectively fulfilling workers' knowledge-needs by preserving, sharing and reusing task-relevant knowledge is essential for realizing knowledge management and promoting business intelligence. Organizations can provide task-relevant knowledge through knowledge flows (KF), which represent the flow of an individual or group's knowledge-needs and referencing behavior of codified knowledge during task execution.

Numerous recent studies have focused on KF models and applications in business and academic contexts. One major research theme focuses on knowledge sharing among knowledge workers. For example, researchers cite prior studies and propose new ideas through publishing papers, thereby creating KFs in the realm of science [1]; and in the business domain, KFs facilitate knowledge sharing during the execution of tasks [2]. By analyzing workers' knowledge-needs, KFs can be discovered, and used to recommend appropriate codified knowledge [3].

C. Huemer and P. Lops (Eds.): EC-Web 2013, LNBIP 152, pp. 80–87, 2013.

When a task involves teamwork, knowledge workers have different roles and task functions, so they usually have diverse knowledge-needs. However, conventional KF models do not provide different KF perspectives to fulfill team members' diverse needs. Although several KF models have been proposed, they do not consider the concept of virtual KFs. Our previous work [4] proposed a KF view model for the construction of virtual KFs to serve workers' knowledge-needs. Virtual KFs are derived from a KF, and provide abstracted knowledge for different roles.

However, our prior work is generally a manual, expert-based method. Moreover, the links between KF views and codified knowledge objects are missing, and workers will not know where to access concrete knowledge documents. Hence, we revise the KF view model, and present a text mining approach to deriving KF views. Generally, codified knowledge objects, such as documents that include semantic content, are exploited. Text similarity measures are employed to estimate knowledge demands for different roles. Then, concept distributions in different tasks are used to identify descriptive topics for representing virtual knowledge nodes. This work contributes to a comprehensive KF view model and data-driven algorithms for generating KF views.

2 Related Work

Knowledge flow (KF) research focuses on how KFs transmit, share and accumulate knowledge in a team. KFs reflect the level of knowledge cooperation between workers or processes, and influence the effectiveness of teamwork or workflow [2]. Sarnikar and Zhao [5] developed a knowledge workflow framework to automate KFs across an organization by integrating workflow and knowledge discovery techniques. Luo et al. [6] designed a textual KF model for a semantic link network. They can recommend appropriate browsing paths to users after evaluating their interests and inputs. KFs also express the sequence of information-needs and knowledge reference patterns when workers perform tasks. Lai and Liu [3] constructed time-ordered KFs from document access logs for modeling workers' knowledge referencing behavior and recommending task-relevant knowledge to workers.

Workflow technology supports the management of organizational processes. Due to the increasing complexity of workflows and the variety of participants, there is a growing demand for flexible workflow models capable of providing appropriate process abstractions [7-9]. Our previous works [7] generated process-views, virtual processes, by an order-preserving approach to preserve the original order of tasks in a base process. A role-based method [10] was also proposed to discover role-relevant process-views for different workers. We further applied the process-view model for KFs to construct KF views [4]. However, our prior studies required expert involvement. Flow designers had to select seed nodes to be concealed to derive virtual nodes. Some profiles and parameters had to be manually specified to evaluate role-task relevance. Moreover, maintaining a specific ontology for the generalization of knowledge nodes is labor-intensive and unfeasible in the rapidly changing business environment.

3 Modeling Virtual Knowledge Flows

3.1 Knowledge Flow and Knowledge-Flow Views

A KF that may have multiple KF views is referred to herein as a *base* KF. A KF view is generated from either base KFs or other KF views, and is considered a virtual KF. Fig. 1 illustrates knowledge sharing based on KF views. Assume that the base KF shown in Fig. 1 is the KF of a software development workflow. Marketers do not need to know every concept in the KF, although they must know software quality topics in order to better serve customers. An appropriate KF view can be derived for the sales representatives as follows: kn_1 to kn_3 are mapped into vkn_1; kn_4 and kn_5 are mapped into vkn_2; kn_6 and kn_7 are mapped into vkn_3. KF views present codified knowledge at suitable granularity; thus, different participants can have their own KF views serving their individual needs.

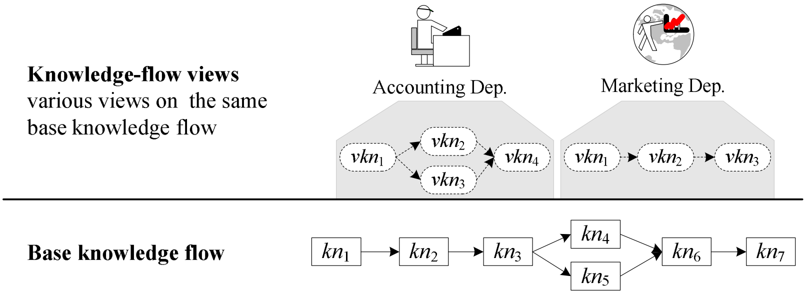

Fig. 1. Illustrative examples of knowledge-flow views

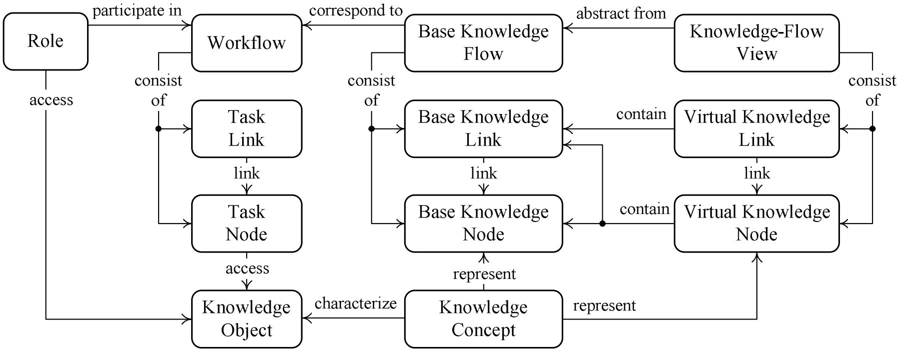

Fig. 2. Knowledge-flow view model

Fig. 2 illustrates how the components of our model are related. To reflect the progress of knowledge needs from the workflow aspect, a KF/knowledge node/knowledge link corresponds to a workflow/task node/task link. *Knowledge concepts*, i.e., key features that characterize codified knowledge objects, included in a knowledge node are the knowledge required by workers to fulfill the corresponding task (node). For example, a document about usability study may be accessed by task node “Web testing”, thus the corresponding knowledge node may include a concept “user experience”, a representative topic of the document. Furthermore, a KF view has a corresponding base KF from which it is derived. Generally, a virtual knowledge node is an abstraction of a set of base knowledge nodes and links.

3.2 Formal Definitions

Definition 1 (workflow): A *workflow WF* is a 2-tuple $\langle TN, TL\rangle$, where

1. *TN* is a set of *task nodes*. Each task node may access a set of knowledge objects.
2. *TL* is a set of *task links*. A task link is denoted by $t\text{-}link(tn_x, tn_y)$ to indicate that the routing can proceed from task node tn_x to tn_y. Links $t\text{-}link(tn_x,\varnothing)$ and $t\text{-}link(\varnothing, tn_y)$ denote that tn_x and tn_y are start and end nodes, respectively.
3. *Path*, *adjacent*, and *ordering relation*: A *path* is a sequence of task links. Two distinct task nodes tn_x and tn_y are *adjacent* if $t\text{-}link(tn_x, tn_y)$ or $t\text{-}link(tn_y, tn_x)$ exists. For tn_x, $tn_y \in TN$: (a) If there is a path from tn_x to tn_y, then the ordering of tn_x is higher than y, i.e., tn_x precedes tn_y. Their ordering relation is denoted by $tn_x > tn_y$ or $tn_y < tn_x$. (b) If no path exists from tn_x to tn_y or from tn_y to tn_x, then tn_x and tn_y are ordering independent, denoted by $tn_x \infty tn_y$, i.e., tn_x and tn_y proceed independently.
4. Knowledge objects represent organizational codified knowledge such as documents and databases. Knowledge concepts signify topics or keywords that characterize their corresponding knowledge objects. That is, a set of knowledge objects $KO_x=\{ko_1, \ldots, ko_m\}$ accessed by task node tn_x are characterized by a set of knowledge concepts $KC_x=\{kc_1, \ldots, kc_n\}$.
5. An organizational *role*, i.e., an abstraction of workers, is represented by a set of knowledge objects to indicate its required background knowledge or experience.

Definition 2 (knowledge flow): A *knowledge flow KF* is a 2-tuple $\langle KN, KL\rangle$, where

1. *KN* is a set of *knowledge nodes*. A knowledge node kn_x contains a set of knowledge concepts extracted from their corresponding knowledge objects, i.e., kn_x = {knowledge concept kc_i | $kc_i \in KC_x'$, $KC_x' \subseteq KC_x$, and KC_x is the set of concepts extracted from the knowledge objects (KO_x) accessed by task node tn_x}.
2. *KL* is a set of *knowledge links*. A knowledge link, denoted by $k\text{-}link(kn_x, kn_y)$, indicates that knowledge access proceeds from knowledge node kn_x to kn_y. The definitions of path, adjacent, and ordering relation in the knowledge flow are similar to those in workflow, and are omitted for brevity.

Definition 3 (knowledge-flow view). A *knowledge-flow view KFV* is a 2-tuple $\langle VKN, VKL\rangle$, where *VKN* is a set of *virtual knowledge nodes* and *VKL* is a set of *virtual knowledge links*.

According to the different properties of a KF, various methods can be developed to derive a KF view. Since task execution sequence (i.e. sequence of knowledge access) is a crucial property for business applications and analysis, our previous work, an *order-preserving* abstraction approach [7], is adopted to generate KF views. Intuitively, a virtual knowledge node/link is an aggregation of a set of base knowledge nodes/links. The approach ensures that the original knowledge access order revealed in a base KF is preserved. Namely, the ordering relation between two virtual knowledge nodes held in a KF view infers that the ordering relations between the respective members of these virtual activities hold in the base KF. The formal definition is described below. The proof of KF view order preservation is similar to that for process-view [7], and is omitted. Cyclic cases are also referred to [7]. In addition, task boundaries are utilized while the derivation of knowledge concepts for KF views (cf. Section 4.3). Therefore, the derived knowledge-flow view is *workflow-aware*, since task boundary and execution sequence are considered during the abstraction process.

Definition 4 (order-preserving knowledge-flow view): Given a knowledge flow $KF = \langle KN, KL \rangle$, a *knowledge-flow view KFV* is a 2-tuple $\langle VKN, VKL \rangle$, where

1. *VKN* is a set of *virtual knowledge nodes*. A virtual knowledge node vkn_x is a 3-tuple $\langle KN_x, KL_x, KC_x \rangle$, where
   (a) Members of KN_x are knowledge nodes or previously defined virtual knowledge nodes. For any $kn_i \in KN$, $kn_i \notin KN_x$, ordering relation $\Re \in \{<, >, \infty\}$: if $\exists\, kn_j \in KN_x$ such that $kn_i \,\Re\, kn_j$ holds in *KF*, then $kn_i \,\Re\, kn_k$ holds in *KF* for all $kn_k \in KN_x$. This means that the ordering relations between kn_i and all members (base knowledge nodes) of KN_x are identical in *KF*.
   (b) $KL_x = \{k\text{-}link(kn_i, kn_j) \mid kn_i, kn_j \in KN_x$ and $k\text{-}link(kn_i, kn_j) \in KL\}$.
   (c) $KC_x = \{$knowledge concept $c_j \mid c_j \in \cup KC_i'$, $\forall kn_i \in KN_x$, $KC_i' \subseteq KC_i$, and KC_i is the set of concepts associated with knowledge node $kn_i\}$.
2. *VKL* is a set of *virtual knowledge links*. A virtual knowledge link from vkn_x to vkn_y, denoted by $vk\text{-}link(vkn_x, vkn_y)$, exists if $k\text{-}link(kn_i, kn_j)$ exists, where kn_i is a member of vkn_x, and kn_j is a member of vkn_y.

4 Discovering Virtual Knowledge Flows

Based on the above definitions, this section describes the procedure and algorithms for discovering KF views.

4.1 Estimating Knowledge Demands

Knowledge needs are subjective, and can be obtained from explicit user profiles or from implicit search and browsing logs. As an exploratory study, we simply utilize text similarity as the base for estimating knowledge demands. The basic idea is inspired by novelty-based recommendation. As shown in Fig. 2 and Definition 1, each role is associated with a set of knowledge objects as background knowledge or experience. Thus, a role is signified by its associated knowledge objects. Knowledge objects of knowledge nodes are less *understandable* to a role if they are less similar to the role profile, and vice versa. The more unfamiliar knowledge nodes must be abstracted to provide more general concepts in order to enhance knowledge comprehensibility and sharing. Without loss of generality, we may use documents to represent knowledge objects.

The vector space model has been applied in many content-based recommendation systems and information retrieval applications. Features (terms) of knowledge objects are extracted after stop-word removal, stemming and term weighting. Each codified knowledge object is described by a term vector comprised of representative terms and their term weights. We employ the well-known *tf-idf* approach to calculate term weights. The weight of term t_i in document d_j is $w_{ij} = tf_{ij} \times \log(F_D/f_{Di})$, where tf_{ij} denotes the term frequency of term t_i in document d_j; f_{Di} is the number of documents that contain the specific term t_i; and F_D is the total number of documents. The similarity between documents is usually measured by the cosine similarity measure. Two documents are considered similar if the cosine similarity score is high. The cosine similarity of two documents, d_1 and d_2, is $sim(d_1, d_2) = \vec{d}_1 \bullet \vec{d}_2 / |\vec{d}_1| \bullet |\vec{d}_2|$, where $\vec{d}_1$ and $\vec{d}_2$ are the feature vectors of d_1 and d_2, respectively.

The understandability degree of a knowledge node with regard to a role is estimated according to the text similarity of knowledge objects. We use the average similarity between knowledge nodes and roles as the understandability value. That is, $und(kn_x,r) = avg(sim(d_m^{kn_x}, d_n^r))$, $\forall d_m^{kn_x} \in kn_x, \forall d_n^r \in r$.

4.2 Generating the Knowledge-Flow View Structure

Based on the degrees of understandability, roles' KF views can be derived. Algorithm 1 determines the set of virtual knowledge nodes of a KF view from a base KF. The process begins with the *highest ordering* nodes in the base KF (line 8). When the *total understandability degree* of a set of base nodes approximates the *granular threshold TH*, a virtual node is found (line 9). Total understandability degree is the sum of understandability degrees of a set of base nodes. Namely, $und(KN, r) = \sum und(kn_x, r)$ $\forall kn_x \in KN$. Granular threshold *TH* determines the granularity of generated KF views. When the sum of the understandability degrees of some base nodes approximates the threshold value, these nodes can form a virtual node, which is deemed to be sufficiently understandable to the role. A larger *TH* corresponds to the generation of fewer virtual nodes (and more base nodes included in a virtual node). The above steps are repeated against residual base nodes until virtual nodes cover all the base nodes of the base KF. Thus, the virtual knowledge node set of the target KF view is found.

Algorithm 1 (The generation of virtual knowledge node set)

```
1:  input: a base knowledge flow $BKF = \langle BKN, BKL \rangle$; $und(kn_x) \le TH, \forall kn_x \in BKN$
2:  output: the set of virtual knowledge node ($VKN$) of a KF view $VKF = \langle VKN, VKL \rangle$
3:  begin
4:     $i \leftarrow 1, VKN \leftarrow \varnothing$
5:     repeat
6:        $vkn_i = \langle KN_i, KL_i \rangle \leftarrow \langle \varnothing, \varnothing \rangle$
7:        residual knowledge node set $RKN \leftarrow BKN - \{kn_x \mid \exists\, vkn_i$ s.t. $kn_x \in vkn_i\}$
8:        select a highest ordering node $kn_x$ from $RKN$
9:        $vkn_i \leftarrow$ getVirtualNode($kn_x$, $RKN$, $BKF$)
10:       $VKN \leftarrow VKN \cup \{vkn_i\}$,  $i \leftarrow i + 1$
11:    until $\forall kn_x \in BKN, \exists\, vkn_i$ s.t. $kn_x \in vkn_i$
12:    return $VKN$
13: end
```

Algorithm 2 (genVirtualNode) discovers a virtual knowledge node. Initially, *KN* contains only the given base node kn_x (line 3). *KN* is updated during the *while* loop (lines 7~15) by adding the adjacent nodes that cause *KN* to satisfy three conditions: the order-preserving property (line 10, cf. [7]); the threshold of total understandability degree; and that it does not overlap with previously derived virtual activities. The *repeat-until* loop (lines 4~16) continues until no other adjacent nodes are added to *KN*.

Algorithm 2 (The generation of a virtual node)

```
1:  getVirtualNode(seed node $kn_s$, residual knowledge node set $RKN$, $BKF=\langle BKN,BKL\rangle$)
2:    begin
3:      $vkn = \langle KN, KL\rangle \leftarrow \langle\{kn_s\}, \varnothing\rangle$
4:      repeat
5:        temp knowledge node set $TKN \leftarrow KN$
6:        adjacent node set $AKN \leftarrow \{kn_x | kn_x, kn_y \in RKN, kn_x \notin KN, kn_y \in KN, k\text{-}link(kn_x,kn_y)$ or
          $k\text{-}link(kn_y, kn_x) \in BKL\}$
7:        while $AKN$ is not empty do
8:          select an base node $kn_x$ from $AKN$
9:          remove $kn_x$ from $AKN$
10:         $KN_{\mathrm{tmp}} \leftarrow$ getOrderPreserVN($KN \cup \{kn_x\}, BKF$) //generate order-preserving virtual node
11:         if ( $und(KN_{\mathrm{tmp}}) \le TH$ ) and ($KN_{tmp} \subseteq RKN$ ) then //check threshold
12:           $KN \leftarrow KN_{\mathrm{tmp}}$
13:           $AKN \leftarrow AKN - \{kn_y | kn_y \in AKN \cap KN\}$
14:         end if
15:       end while
16:     until $KN = TKN$
17:     link set $KL \leftarrow \{ k\text{-}link(kn_x, kn_y) | kn_x, kn_y \in KN$, and $k\text{-}link(kn_x, kn_y) \in BKL\}$
18:     return $vkn = \langle KN, KL\rangle$
19:   end
```

4.3 Generating Knowledge-Flow View Content

Finally, knowledge concepts are derived to represent (virtual) knowledge nodes. As described in Section 3.1, a knowledge node corresponds to a collection of knowledge objects that are accessed by the corresponding task node. Knowledge concepts of a knowledge node are the representative topical words generated from the corresponding knowledge objects.

Whether a word/phrase is an appropriate topic for a knowledge node is determined from a single knowledge node and the whole KF aspects. For example, “JUnit” is better than “test case” to represent “unit testing” in a software testing KF. That is, task (knowledge node) boundaries are the curtail factor for selecting suitable keywords from knowledge objects. Therefore, term statistics of inter- and intra-knowledge nodes are used to identify representative knowledge concepts.

First, in order to increase the comprehensibility of KFs and views, documents are mapped to Wikipedia concepts. That is, only Wikipedia terms in the text are recognized as candidates. Wikipedia dumps are utilized for the term extraction.

Next, term distributions of intra-knowledge nodes are measured by *term frequency* (*tf*). Moreover, term statistics of inter-knowledge nodes are measured by *inverse node frequency* (*inf*). Term t_i’s $inf = \log(F_N/f_{Ni})$, where F_N is the number of knowledge nodes, and f_{Ni} is the number of knowledge nodes that contain term t_i. Finally, candidate terms are ranked according to *tf-inf*: *tf-inf* of term t_i in node n_j is $w_{ij} = tf_{ij} \times \log(F_N/f_{Ni})$, where tf_{ij} is the frequency of term w_i in node n_j. The main difference between base and virtual knowledge nodes is the boundary of nodes, and thus the knowledge concept generation process is the same for KF and KF views.

5 Conclusions

This work presents a KF view model for knowledge sharing and navigation. Knowledge granularity of KFs is adapted to the needs of workflow participants. Workers can thus obtain helpful views of a large and complex KF. To support the discovery of role-relevant KF views, this work utilizes text similarity of knowledge objects to measure the degrees of understandability between roles and knowledge nodes. Task execution sequence and knowledge access order are preserved while generating abstracted KFs. Moreover, task boundaries are employed to derive representative knowledge concepts for knowledge nodes. Therefore, role-relevant KF views are automatically generated using the proposed algorithms. Accordingly, knowledge management systems can disseminate KFs at suitable granularities for various organizational roles.

Acknowledgements. This research was supported by the National Science Council of the Taiwan under Grant No. NSC 99-2410-H-009-034-MY3 and NSC 102-2811-H-009-002.

References

1. Zhuge, H.: Discovery of Knowledge Flow in Science. Communications of the ACM 49, 101–107 (2006)
2. Zhuge, H., Guo, W.: Virtual Knowledge Service Market - for Effective Knowledge Flow within Knowledge Grid. Journal of Systems and Software 80, 1833–1842 (2007)
3. Lai, C.-H., Liu, D.-R.: Integrating Knowledge Flow Mining and Collaborative Filtering to Support Document Recommendation. Journal of Systems and Software 82, 2023–2037 (2009)
4. Liu, D.-R., Lin, C.-W., Chen, H.-F.: Discovering Role-Based Virtual Knowledge Flows for Organizational Knowledge Support. Decision Support Systems 55, 12–30 (2013)
5. Sarnikar, S., Zhao, J.: Pattern-Based Knowledge Workflow Automation: Concepts and Issues. Information Systems and E-Business Management 6, 385–402 (2008)
6. Luo, X., Hu, Q., Xu, W., Yu, Z.: Discovery of Textual Knowledge Flow Based on the Management of Knowledge Maps. Concurrency and Computation: Practice and Experience 20, 1791–1806 (2008)
7. Liu, D.-R., Shen, M.: Workflow Modeling for Virtual Processes: An Order-Preserving Process-View Approach. Information Systems 28, 505–532 (2003)
8. Weidlich, M., Smirnov, S., Wiggert, C., Weske, M.: Flexab - Flexible Business Process Model Abstraction. In: The Forum of the 23rd Intl. Conf. on Advanced Information Systems Engineering (CAiSE Forum 2011), pp. 17–24. CEUR-WS.org (2011)
9. Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Abstractions in Process Mining: A Taxonomy of Patterns. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 159–175. Springer, Heidelberg (2009)
10. Shen, M., Liu, D.-R.: Discovering Role-Relevant Process-Views for Disseminating Process Knowledge. Expert Systems with Applications 26, 301–310 (2004)