

# Facilitating Cooperative Knowledge Support by Role-Based Knowledge-Flow Views

Chih-Wei Lin, Duen-Ren Liu, and Hui-Fang Chen

**Abstract**—Effective knowledge support relies on providing operation-relevant knowledge to workers promptly and accurately. A knowledge flow represents an individual's or a group's knowledge-needs and referencing behavior of codified knowledge during operation performance. The flow has been utilized to facilitate organizational knowledge support by illustrating workers' knowledge-needs systematically and precisely. However, conventional knowledge-flow models cannot work well in cooperative teams, which team members usually have diverse knowledge-needs in terms of roles. The reason is that those models only provide one single view to all participants and do not reflect individual knowledge-needs in flows. Hence, we propose a role-based knowledge-flow view model in this work. The model builds knowledge-flow views (or virtual knowledge flows) by creating appropriate virtual knowledge nodes and generalizing knowledge concepts to required concept levels. The customized views could represent individual role's knowledge-needs in teamwork context. The novel model indicates knowledge-needs in condensed representation from a roles perspective and enhances the efficiency of cooperative knowledge support in organizations.

**Keywords**—cooperative knowledge support, knowledge flow, knowledge-flow view, role-based models

## I. INTRODUCTION

WORKING in knowledge economy era, workers need a lot of knowledge and documents to perform operations. Thus, effectively and accurately fulfilling workers' knowledge-needs is critical to build knowledge support systems. The knowledge-needs are arising from the gaps between workers' knowledge and the necessary requirements of tasks [1]. They would be represented by knowledge concepts and converged in knowledge nodes. Knowledge flows are composed of knowledge nodes to illustrate the sequences of individuals' or groups' knowledge-needs and referencing behavior of codified knowledge when workers perform operations [2].

Many organizations exploit knowledge support systems to provide operation-relevant knowledge to improve workers' productivity. Because the way to represent knowledge-needs is essential to setup such systems, research in formulating knowledge-needs and streamlining knowledge provision has gathered importance recently [3-5].

Numerous studies were carried out on the knowledge-flow models and their applications in business and scientific research contexts [6-10]. Zhuge [9] indicated that knowledge sharing among researchers through paper citations could form

knowledge flows. In addition, knowledge flows would facilitate knowledge support in workflow execution by transmitting knowledge between actors [11]. Lai and Liu [2] discovered knowledge flows by analyzing workers' document access logs and recommend appropriate codified knowledge to them based on the knowledge flows.

Nevertheless, workers in cooperative teams usually have different knowledge-needs by their roles or job characteristics. Conventional knowledge-flow models cannot work well in such teamwork environments because they only provide one single view to all team members without considering individual difference. So, we proposed a knowledge-flow view (KfV) model to deal with the issue in our previous work [12]. In KfV model, we apply order-preserving approach and minimum generalization policy to abstract knowledge nodes of a base knowledge flow to generate corresponding KfVs based on job characteristics, for example, concealing confidential information to comply with organizational security rules. Hence, each team member has an appropriately KfV to illustrate his or her knowledge-needs.

However, the preliminary KfV model we proposed before does not consider workers' roles which they play in teams. In fact, tasks are assigned to dedicated roles based on the characteristics of tasks to ensure quality and security. Therefore, workers have different knowledge-needs in terms of roles. For example, in a mobile phone development project, product engineers are responsible for product development; yet, marketing analyzers are responsible for promoting products. So, the product engineers need knowledge with finer granularity to solve technical problems; but, the marketing analyzers only need such knowledge with coarser granularity for communication purpose. That is, product engineers and marketing analyzers have different needs relating to the same knowledge due to their roles. Consequently, building knowledge-flow views from a roles perspective is necessary to illustrate knowledge-needs properly in teamwork environments.

In this work, we propose a role-based knowledge-flow view (r-KfV) model to illustrate team participants' knowledge-needs from a roles perspective. We first setup operation profiles to calculate relevance degrees between roles and knowledge nodes. The relevance degrees are the criteria to decide how to generate virtual knowledge nodes which comprise some knowledge nodes of a base knowledge flow. A virtual knowledge node should include relevant enough knowledge nodes only and have order-preserving property. In addition, roles may require different granular knowledge, such as general or specific ones, to conduct their operations. The granularity requirements are described by knowledge degree profiles. Based on the knowledge degrees profiles, a concept

Authors are with National Chiao Tung University, No. 1001 Ta Hseuh Rd., Hsinchu, Taiwan (phone:+886 3 5712121-57401; fax:+886 3 5723792; e-mail: hjwlin.iim93g@nctu.edu.tw, dliu@iim.nctu.edu.tw, hfchen@iim.nctu.edu.tw).

abstraction method is designed to obtain proper knowledge through referring domain ontologies. Finally, the r-KFVs can be discovered for all roles.

The novel r-KFV model advances the theoretical scope of knowledge flow research in teamwork environments. It also improves the effectiveness of cooperative knowledge support in organizations.

The rest of this paper is organized as follows: Section 2 provides a summary of related research. Section 3 briefly explains the preliminary KFV model. In Section 4, we bring roles into KFV model and illustrate the concepts and framework of the r-KFV model. The methods to generate r-KFVs are introduced in Section 5. Conclusions are finally made in Section 6.

## II. BACKGROUND AND RELATED WORK

### A. Knowledge Support

From resource-based view of knowledge management, knowledge is one key asset to help organizations maintain competitive advantages [13]. Workers need task-relevant knowledge to perform their tasks in a knowledge-intensive working environment. For example, the KnowMore system [14] established a context-aware knowledge support platform to deliver proper knowledge to workers based on task descriptions and operation contexts. Liu and Wu [5] designed a K-support system provides knowledge support to fulfill workers' information-needs by analyzing their access behavior and relevance feedbacks on recommended documents. Obviously, effective knowledge support is critical for organizations to conduct knowledge management and improve employee productivity.

### B. Knowledge Flow

Knowledge flows would transmit, share, and accumulate knowledge while workers pass knowledge from one person to another person during performing tasks. They would reflect the effectiveness of knowledge sharing and impact the performance of teamwork [10]. For example, a knowledge flow can gather knowledge from one member and carry it to others, thereby facilitating knowledge sharing in teams. Several formal models have been proposed to define knowledge flow. Kim, Hwang, and Suh [6] proposed a knowledge support model that combines knowledge flows and a process-oriented approach to capture, store, and transfer knowledge. Lai and Liu [2] constructed a time-ordered knowledge-flow model to determine the sequence of workers' knowledge referencing behavior. Zhao and Dai [15] integrated business processes and knowledge flows based on RAD (role-activity-diagram) model. Those knowledge-flow models and applications increase the effectiveness of knowledge sharing and knowledge support in organizations.

### C. Process Abstraction

Business process management allows enterprises to analyze, simulate, design, implement, control and monitor their overall

business processes. In practice, participants involved in a business process need a flexible view that is capable of providing appropriate process information. Liu and Shen [16] presented an order-preserving approach for generating process views. They are virtual processes derived from a base process to provide abstracted process information to each participant. Eshuis and Grefen [17] suggested a two-step approach to construct process views on structured process models by concealing confidential activities and omitting unnecessary activities. Moreover, Shen and Liu [18] suggested a role-based approach to discover role-relevant process views to serve different roles in workflow. The approach generates process views automatically based on the relevance degrees between participating roles and operations. This work adapts the order-preserving approach and the relevance degrees concept to generate role-based knowledge-flow views.

### D. Role-based Access Control

Role-based access control (RBAC) [19] is widely used as an information security control mechanism. In RBAC, access permissions are associated with roles and users are assigned to appropriate roles to reduce the complexity of access control processes. For example, Li, Du, and Wong [20] exploited RBAC mechanism to address the conflicts of access controls among collaborative applications such as workflow and supply chain. The RBAC mechanism is used to manage the information flow among those applications to balance the minimum privilege of accessing operation-relevant knowledge and the maximum privilege of sharing general knowledge.

## III. PRELIMINARY KNOWLEDGE-FLOW VIEW MODEL

A preliminary knowledge-flow view (KFV) model was proposed in our previous work [12] to generate knowledge-flow view from a base knowledge flow. A base knowledge flow represents an individual's or a group's knowledge-needs and referencing behavior of codified knowledge while performing operations. We use a mobile phone development process shown in Fig. 1 to explain how to discover the corresponding base knowledge flow from the process.

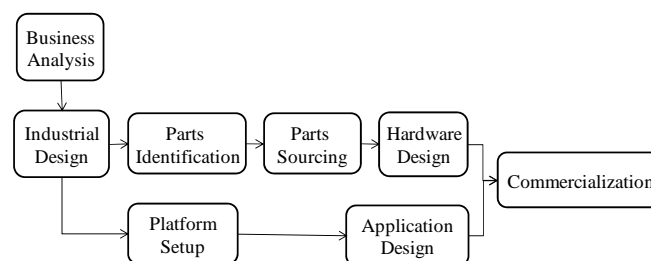


Fig. 1 A mobile phone development process

According to the process, a knowledge-flow designer may either consult domain experts or investigate team participants' document access logs to identify team members' knowledge-needs. For example, the team members may have

the knowledge-needs of “geographic segmentation” and has to access “consumer behavior” related documents while conducting the “business analysis” operation. So, the designer would put those knowledge concepts into a knowledge node to represent the knowledge-needs about the “business analysis” operation. Finally, the designer collects all knowledge-needs, formulates the knowledge-needs into knowledge concepts based on domain ontologies, generates knowledge nodes, and forms the base knowledge flow as shown in Fig. 2.

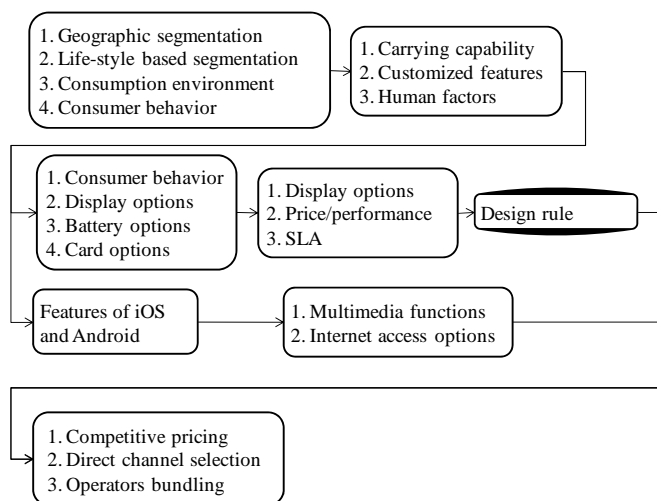


Fig. 2 A base knowledge flow

A base knowledge flow may have multiple knowledge-flow views (KFVs) to serve different participants. The KfV, an abstracted form of a base knowledge flow, reveals abstracted knowledge concepts by operation or user characteristics. Fig. 3 shows the relationship between a base knowledge flow and its KFVs.

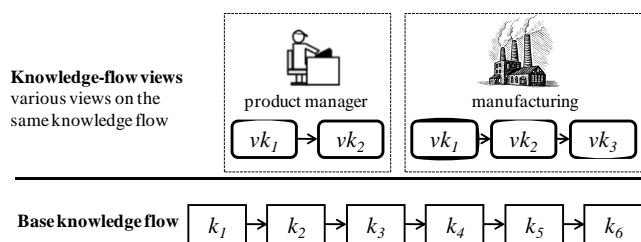


Fig. 3 Relationship between a base knowledge flow and KFVs

Supposing the base knowledge flow in Fig.3 is a manufacturing process. Product managers need not know all knowledge concepts in detail. However, they need general manufacturing knowledge to increase communication effectiveness while discussing the trends of product yield with manufacturing staff. Hence, it is helpful for product managers if the knowledge-flow designer generates a KfV for them. For instance, the KfV of product managers could abstract knowledge nodes  $k_1$  and  $k_2$  to a virtual knowledge node  $vk_1$  and abstract  $k_3, k_4, k_5,$  and  $k_6$  to  $vk_2$ . Moreover, manufacturers also have their own KfV which contains specific manufacturing knowledge (represented in  $vk_2$ ) and general knowledge of other

domains (represented in  $vk_1$  and  $vk_3$  respectively). Consequently, process participants have their customized KFVs to represent their knowledge-needs in collaborative environments.

The following summarizes the basic definitions of the base knowledge flow mode and the KfV model.

**Definition 1 (Domain Ontology, O):** Domain ontology is constructed to define knowledge concepts and their hierarchical relationships in a specific domain. A domain ontology is defined as  $O = \langle C, HR \rangle$ , where  $C$  is a set of knowledge concepts and  $HR = \{hr \mid hr \in C \times C\}$ .  $HR$  is a set of hierarchical relations that define the parent-child relationships between two knowledge concepts in  $C$ . Given two knowledge concepts  $x$  and  $y$ , if  $x$  has a downward link to  $y$  (or  $y$  has an upward link to  $x$ ) in  $O$ , then  $x$  is the parent concept of  $y$  and  $y$  is the child concept of  $x$ . Two semantic relations, Generalization and Specialization, are used to describe the parent-child relations.  $Specialization(x) = \{y \mid y \text{ is a child concept of } x\}$  and  $Generalization(y) = \{x \mid x \text{ is a parent concept of } y\}$ .

**Definition 2 (Concept Level, CL):**  $CL$  is a measure of granularity of knowledge concepts. The concept level  $cl$  of a knowledge concept  $c$  in a domain ontology  $o$  can be denoted as  $CL(o.c)$ . We assign 1.0 to  $cl$  for the root of  $o$ . If  $cl$  of a knowledge concept  $c$  is  $d$ , then the concept levels of its child concepts are  $d+1$ . Obviously, a knowledge concept with a larger (smaller)  $cl$  contains more specific (general) knowledge.

**Definition 3 (Base Knowledge Flow, BKF):** A base knowledge flow (BKF) is a 2-tuple  $\langle BKN, BD \rangle$ , where

1.  $BKN$  is a set of knowledge nodes. Each knowledge node consists of a set of knowledge concepts. The knowledge concepts of a knowledge node  $x$  are denoted by  $Concept(x) = \{o_i.c_j \mid \text{knowledge concept } c_j \text{ can be identified in domain ontology } o_i\}$ .
2.  $BD$  is a set of dependencies. A dependency is denoted by  $dep(x, y)$ , where  $x$  and  $y$  are knowledge nodes in  $BKN$ . This notation indicates that workers refer the knowledge concepts of  $x$  before refer the knowledge concepts of  $y$ .

**Definition 4 (Knowledge-Flow View, KfV):** A knowledge-flow view  $KfV$ , or a virtual knowledge flow, is an abstraction form of a base knowledge flow and is denoted by  $KfV \langle VKN, VD \rangle$ , where  $VKN$  is a set of virtual knowledge nodes and  $VD$  is a set of virtual dependencies.

**Definition 5 (Virtual Knowledge Node):** Given a base knowledge flow  $BKF = \langle BKN, BD \rangle$ , a virtual knowledge node  $vx$  consists of a set of member knowledge nodes  $x_1, x_2, x_3 \dots x_n$  belonging to  $BKN$ . The knowledge concepts of  $vx$  are derived from the member knowledge nodes' knowledge concepts by a knowledge concept abstraction method. Thus, A virtual knowledge node  $vx$  can be denoted by a 2-tuples  $\langle KN, KC \rangle$ , where  $KN$  (Knowledge Node Set) is a set of member knowledge nodes and  $KC$  (Knowledge Concept Set) is a set of knowledge

concepts.

**Definition 6 (Virtual Dependency):** given a  $BKF = \langle BKN, BD \rangle$ , a corresponding  $KFV = \langle VKN, VD \rangle$  and two virtual knowledge nodes  $vx$  and  $vy$  in  $VKN$ , a virtual dependency  $vdep(vx, vy)$  from  $vx$  to  $vy$  exists if  $dep(x, y)$  is in  $BD$ , where  $x$  is a member knowledge node of  $vx$  and  $y$  is a member knowledge node of  $vy$ . The virtual dependencies connect virtual knowledge nodes to form a  $KFV$  and indicate the referring order of knowledge concepts.

Fig. 4 shows the relationship between the knowledge flow model and the KFV model.

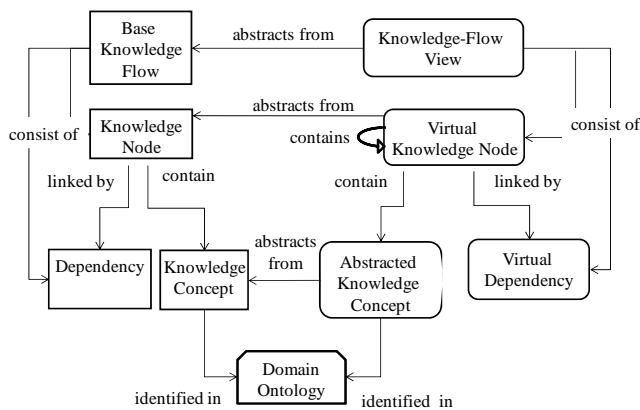


Fig. 4 Relationships between knowledge flow model and KFV model.

#### IV. INTEGRATING ROLES INTO KNOWLEDGE-FLOW VIEW MODEL

We introduce a role-based framework into the knowledge-flow view (KFV) model and present the key concepts of the role-based knowledge-flow view (r-KFV) model in this section.

##### A. Deriving Relevance Degrees between Roles and Knowledge Nodes

A role may refer to some knowledge nodes frequently but refer to others rarely due to its responsibilities and authorities. The closeness of a role to knowledge nodes is distinct and indicates how important the knowledge nodes to the role are. We set relevance degree to represent the level of closeness. According to the relevance degrees, virtual knowledge nodes are generated by abstracting relevant knowledge nodes from a base knowledge flow. That is, a virtual knowledge node is a meaningful knowledge unit and should be relevant enough to the role.

The relevance degrees can be specified by knowledge-flow designers. But, it's time consuming and complex for them to manually determine the degrees for each role. Hence, we calculate relevance degrees by a derivation process. First, whether a role is authorized to perform an operation is determined by organizational security rules which are

expressed as *permission rules* and defined in role-based access control systems [19]. Second, we build *role profiles* to discover the relative levels between roles and authorized operations. The relative level is a measure of role-operation associations and comes from operation logs. For example, role  $r$  performs three authorized operations -  $op_1, op_2, op_3$ . Based on operation logs, role  $r$  performs  $op_1$  twice,  $op_2$  seven times and  $op_3$  once; then the number of times that role  $r$  performs all operations is 10. Hence, the relative level of  $r$  to  $op_1$  is 0.2 (2/10), 0.7 (7/10) for  $op_2$  and 0.1 (1/10) for  $op_3$ .

Next, we build *knowledge node profiles* to store the linkage between knowledge nodes and operations. Since a knowledge node is a collection of knowledge-needs which deriving from one or many operations, a knowledge node profile can illustrate which operations are related to the knowledge node. In the final step of the derivation process, we apply different roles as criteria for evaluating the relevance degrees through the role-base framework which includes permission rules, role profiles, and knowledge node profiles.

Identifying a virtual knowledge node is a repeated loop to merge base knowledge nodes one by one and totaling their relevance degrees during each merge. Once the aggregated relevance degree reaching a certain threshold, a virtual knowledge node is identified. Consequently, knowledge-flow designers can generate appropriate virtual knowledge nodes to form r-KFVs based on the role-based framework.

##### B. Deriving Knowledge Concepts for Virtual Knowledge Nodes

A concept abstraction method is applied to abstract knowledge concepts for a virtual knowledge node by merging its member knowledge nodes. However, the member knowledge nodes only contain the most specific knowledge concepts to represent operation executors' knowledge-needs at the build time of knowledge flow. Those knowledge concepts may not conform to roles' demands from information granularity perspective. Hence, the concept abstraction method abstracts the knowledge concepts to proper concept levels first rather than merging them directly. We exploit concept levels (CLs) to represent the granularity of knowledge concepts. In fact, participants in a team need specific knowledge concepts with the largest  $cl$  to perform operations accurately; they also need other general knowledge concepts with smaller  $cl$  to communicate with other members effectively. Hence, a virtual knowledge node may consist of knowledge concepts with different  $cl$  by roles' requirements.

The concept abstraction method uses operation profiles and knowledge degree profiles to generalize knowledge concepts. *Operation profiles* describe a set of mandatory knowledge concepts in different domain ontologies while performing an operation. *Knowledge degree profiles* represent the required granularity of knowledge concepts for performing a certain operation by a specific role. For simplifying knowledge degree profiles, the knowledge degrees are set by domain ontologies instead of by individual knowledge concepts. Eventually, the concept abstraction method derives appropriate knowledge

concepts with proper  $cl$  to fulfill roles' demands and merges them into a virtual knowledge node.

Fig. 5 shows the role-based framework for measuring the relevance degrees and abstracting knowledge concepts for a virtual knowledge node. The relative level of role  $r_1$  to operation  $op_1$  is 0.6 (shown in role profile); the mandatory knowledge concepts to perform  $op_1$  are  $c_1$  and  $c_2$  in  $o_1$  and  $c_3$  in  $o_2$  (shown in operation profile); the knowledge degree of the knowledge concepts in  $o_2$  which role  $r_1$  need to perform  $op_1$  is 0.8 and 0.2 in  $o_1$  (shown in knowledge degree profile). Moreover,  $op_1$  is associated with knowledge node  $kn_1$  (shown in knowledge node profile) and  $r_1$  is authorized to perform  $op_1$  (shown in permission rule). We define that the relevance degree between  $r_1$  and  $kn_1$  is the maximum of the relative levels of all authorized operations (i.e.  $op_1, op_2$ ) which are associated with  $kn_1$ . So, the relevance degree between  $r_1$  and  $kn_1$  is  $\max(0.6, 0.3) = 0.6$ . Based on the role-based framework, we can generate a virtual knowledge node as well as its knowledge concepts. The detail steps are described in Section 5.

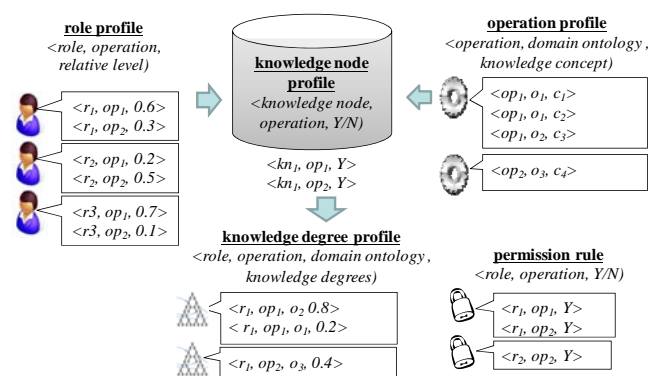


Fig. 5 Role-based framework

## V. DISCOVERING ROLE-BASED KNOWLEDGE-FLOW VIEWS

Generating a role-based knowledge-flow view (r-KFV) from a base knowledge flow involves three steps: (a) generating virtual knowledge nodes, (b) deriving knowledge concepts of virtual knowledge nodes and (c) building virtual dependencies to connect virtual knowledge nodes.

### A. Generating Virtual Knowledge Nodes

The generation procedure starts from taking the highest ordering knowledge node from a base knowledge flow as a seed node. Beginning with the seed, the procedure repeatedly aggregate the adjacent base knowledge nodes of the seed according to the descending order of their relevance degrees. Until the total relevance degree of the aggregated base knowledge nodes approximates a granular threshold  $TH$ , a virtual knowledge node is identified. Other base knowledge nodes which are not aggregated forms a residual knowledge node set ( $RKN$ ). The base knowledge nodes in  $RKN$  mean that

they are not the member knowledge nodes of any virtual knowledge nodes yet. The procedure takes the highest ordering base knowledge node from  $RKN$  as another seed node and repeats another aggregation run. The loop continues until no more base knowledge nodes in  $RKN$ . Hence, all virtual knowledge nodes have been generated.

Moreover, during the aggregating loop, the order-preserving property should be checked. Liu and Shen [16] developed an order-preserving approach that generates process views from a base process in workflow environments. The approach ensures that the implied ordering relations of activities in a process view must comply with the ordering relations of activities in the base process. We adopt the approach to ensure that an r-KFV maintains the knowledge referencing order in the r-KFV as the order in the base knowledge flow.

### B. Deriving Knowledge Concepts of Virtual Knowledge Nodes

Deriving knowledge concepts for a virtual knowledge node is to abstract all knowledge concepts of its member knowledge nodes to proper concept levels based on roles' requirements in terms of knowledge degrees.

Initially, we create an operation set  $OP$  for a virtual knowledge node to record the authorized operations which are associated with its member knowledge nodes by computing knowledge node profiles and permission rules. Then a knowledge degree set  $KD$  is built by converging operation profiles and knowledge degree profiles for each operation in  $OP$ . The elements in  $KD$  are 3-tuples <domain ontology  $o$ , knowledge concept  $kc$ , and knowledge degree  $kdeg$ >, which represent role's requirement in terms of  $kdeg$  about the knowledge concept  $kc$  in the domain ontology  $o$ . Finally, we follow the hierarchy structure of domain ontology  $o$  to generalize knowledge concept  $kc$  to its parent knowledge concept until the concept level of the parent knowledge concept meets roles' requirement  $kdeg$ .

### C. Building Virtual Dependencies

After obtaining all virtual knowledge nodes, the virtual dependencies between them can be determined by the dependencies between their member knowledge nodes. For any pair of virtual knowledge nodes in a r-KFV,  $vx$  and  $vy$ , the virtual dependency  $vdep(vx, vy)$  exists if  $dep(x, y)$  exists in base knowledge flow, where  $x, y$  are knowledge nodes of the base knowledge flow,  $x$  is a member of  $vx$ , and  $y$  is a member of  $vy$ .

### D. An Illustrating Case

According to the development process shown in Fig. 1, we choose sourcing department manager as example role  $r$  to illustrate how to generate a role-based knowledge-flow view. The role  $r$  is responsible for communicating with project manager about project status and appraising the performance of sourcing planners who work in sourcing department and are assigned to the process. The base knowledge flow shown in Fig. 6 includes eight knowledge nodes,  $k_0$  to  $k_7$ . Each knowledge node contains multiple knowledge concepts (shown in rectangles) and has distinct relevance degrees (shown in

parentheses) to role  $r$ .

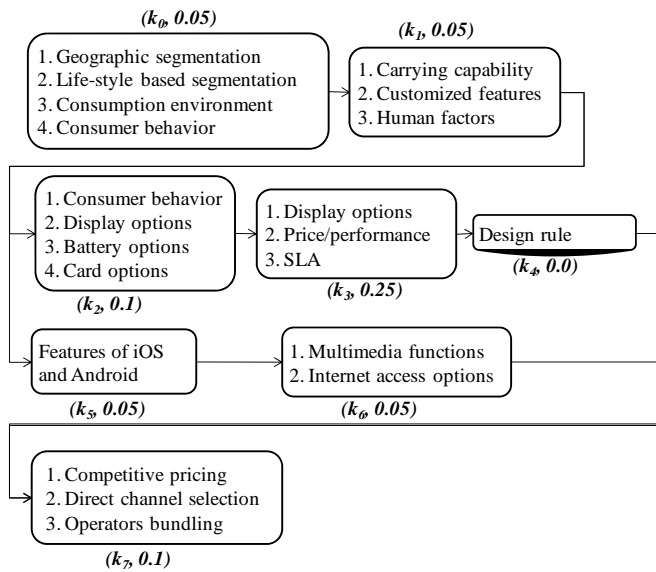


Fig. 6 Example base knowledge flow with relevance degrees.

The next step is to generate virtual knowledge nodes. We assume the threshold,  $TH$ , is set to 0.4 for role  $r$ . Initially, we select a base knowledge node with the highest order,  $k_0$ , as a seed node to generate the first virtual knowledge node  $vk_{n1}$ . After calculating the relevance degrees of member knowledge nodes and checking the order-preserving property, the virtual knowledge node  $vk_{n1}$  should include two member knowledge nodes  $k_0$  and  $k_1$ . Repeating the iteration, the knowledge nodes are merged into four virtual knowledge nodes  $vk_{n1}$ ,  $vk_{n2}$ ,  $vk_{n3}$  and  $vk_{n4}$  respectively. Then, we form an r-KFV shown in Fig. 7 after discovering their virtual dependencies.

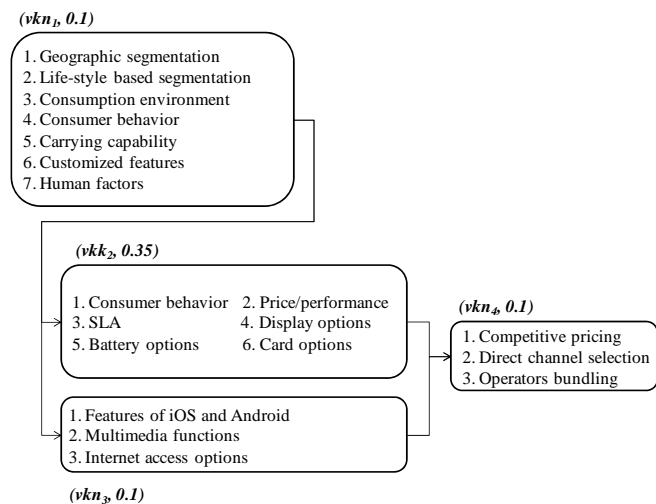


Fig. 7. Role-base knowledge-flow view with  $TH=0.4$ .

The following discussion takes  $vk_{n2}$  as an example to illustrate the concept abstraction method. First, we clarify the domain ontologies related to the process as shown in Fig. 8.

Next, the knowledge-flow designer sets  $r$ 's knowledge-needs in terms of knowledge degrees in role-based framework. Fig. 9 shows the framework and the profiles related to  $r$ .

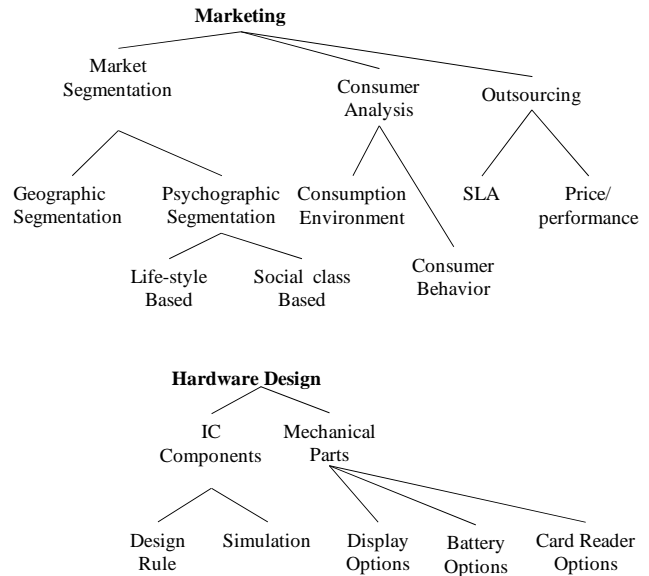


Fig. 8 Marketing domain ontology and hardware design domain ontology

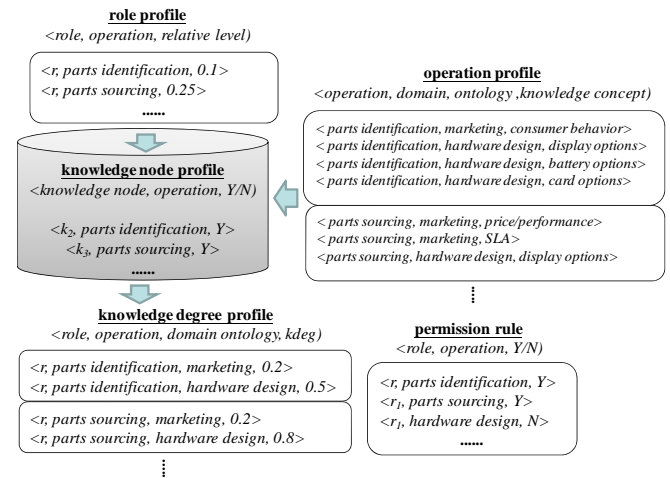


Fig. 9 Role-based framework and related profiles

By mapping operation profile and knowledge degree profile, a knowledge degree set  $KD$  would be built for role  $r$  to represent its requirement. Hence, we can get

$$KD = \{(\text{marketing}, \text{"consumer behavior"}, 0.2), (\text{marketing}, \text{"price/performance of parts"}, 0.2), (\text{marketing}, \text{"SLA"}, 0.2), (\text{hardware design}, \text{"battery options"}, 0.5), (\text{hardware design}, \text{"card options"}, 0.5), (\text{hardware design}, \text{"display options"}, 0.8)\}.$$

Based on the required knowledge degrees and the knowledge concept generalization function, every knowledge concept in *KD* is abstracted to parent concept following domain ontologies until it reach the proper concept level.

Consequently, the knowledge concepts of  $vk_{n_2}$  are “marketing”, “mechanical parts” and “display options”. The same method can apply to other virtual knowledge nodes. Fig. 10 shows the final r-KFV.

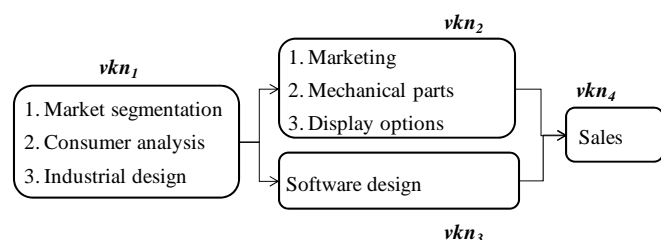


Fig. 10 Final r-KFV

This case shows that r-KFV model can benefit the team members in many ways. First, r-KFVs can give participating roles a full picture of knowledge-needs by showing corresponding knowledge concepts with proper concept levels. Second, team members can describe their knowledge-needs precisely and gain consensus quickly in teams by common domain ontologies and knowledge degree profiles. Last, the knowledge-flow designers can avoid complex and time-consuming tasks to design r-KFVs for participating roles one by one. In summary, the r-KFV model can facilitate organizational knowledge support platforms and help teams improve their communication quality and increase members' productivities.

## VI. CONCLUSIONS

Through knowledge flows, organizations can provide workers with operation-relevant knowledge to meet their knowledge-needs effectively due to knowledge concepts have been identified properly. In teamwork environments, knowledge workers with different roles usually have different knowledge-needs. However, conventional knowledge-flow models do not work well in such context. In this work, the role-based knowledge-flow view (r-KFV) model is proposed to extend the preliminary knowledge-flow view model we suggested in [12]. The r-KFV model examines the knowledge-needs from a roles perspective. An r-KFV comprises a set of virtual knowledge nodes, which are generated from base knowledge nodes according to their relevance degrees to a given role. The knowledge concepts of virtual knowledge nodes are derived by generalizing the knowledge concepts of member knowledge nodes to the appropriate concept levels depending on roles' requirements in terms of knowledge degrees.

The r-KFVs are essential because they provide effective knowledge support for workers involved in knowledge-intensive activities. The novel model enhances the efficiency of conventional knowledge flows, as well as the

effectiveness of knowledge sharing. Currently, we are developing algorithms for generating r-KFVs and develop the system architecture to realize the model. In future, we plan to adapt knowledge-flow views into the context of workflow enacting environments. Ideally, knowledge flows or knowledge-flow views can be utilized as a proactive knowledge support mechanism in accordance with a workflow instance to provide necessary knowledge for workers to conduct tasks. So, it would be interesting to investigate the interactions between knowledge flow (and knowledge-flow view) and business processes to facilitate knowledge dissemination effectively.

## ACKNOWLEDGMENT

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

## REFERENCES

- [1] I. C. Wu, D. R. Liu, and P. C. Chang, "Learning dynamic information needs: a collaborative topic variation inspection approach," *Journal of the American Society for Information Science and Technology*, vol. 60, pp. 2430-2451, 2009.
- [2] C. H. Lai and D. R. Liu, "Integrating knowledge flow mining and collaborative filtering to support document recommendation," *Journal of Systems and Software*, vol. 82, pp. 2023-2037, 2009.
- [3] 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*, vol. 13, pp. 271-284, 2000.
- [4] P. Balatsoukas and P. Demian, "Effects of granularity of search results on the relevance judgment behavior of engineers: Building systems for retrieval and understanding of context," *Journal of the American Society for Information Science and Technology*, vol. 61, pp. 453-467, 2010.
- [5] D. R. Liu and I. C. Wu, "Collaborative relevance assessment for task-based knowledge support," *Decision Support Systems*, vol. 44, pp. 524-543, 2008.
- [6] S. Kim, H. Hwang, and E. Suh, "A process-based approach to knowledge-flow analysis: a case study of a manufacturing firm," *Knowledge and Process Management*, vol. 10, pp. 260-276, 2003.
- [7] D. R. Liu and C. H. Lai, "Mining group-based knowledge flows for sharing task knowledge," *Decision Support Systems*, vol. 50, pp. 370-386, 2011.
- [8] X. Luo, Q. Hu, W. Xu, and Z. Yu, "Discovery of textual knowledge flow based on the management of knowledge maps," *Concurrency and Computation: Practice and Experience*, vol. 20, pp. 1791-1806, 2008.
- [9] H. Zhuge, "Discovery of knowledge flow in science," *Communications of the ACM*, vol. 49, pp. 101-107, 2006.
- [10] H. Zhuge, "Knowledge flow management for distributed team software development," *Knowledge-Based Systems*, vol. 15, pp. 465-471, 2002.
- [11] H. Zhuge, "A knowledge flow model for peer-to-peer team knowledge sharing and management," *Expert Systems with Applications*, vol. 23, pp. 23-30, 2002.
- [12] C. W. Lin, D. R. Liu, and H. F. Chen, "Modeling knowledge-flow view for knowledge support in teamwork," in *Proceedings of the 21st IASTED International Conference on Modelling and Simulation.*, Banff, Alberta, Canada, 2010.
- [13] V. Sambamurthy and M. Subramani, "Special issue on information technologies and knowledge management," *MIS Quarterly*, vol. 29, pp. 1-7, 2005.
- [14] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, and M. Sintek, "Context-aware, proactive delivery of task-specific information: the knowmore project," *Information Systems Frontiers*, vol. 2, pp. 253-276, 2000.

- [15] W. Zhao and W. Dai, "Integrated modeling of business processes and knowledge flow based on RAD," in *IEEE International Symposium on Knowledge Acquisition and Modeling Workshop*, China, 2008, pp. 49-53.
- [16] D. R. Liu and M. Shen, "Workflow modeling for virtual processes: an order-preserving process-view approach," *Information Systems*, vol. 28, pp. 505-532, 2003.
- [17] R. Eshuis and P. Grefen, "Constructing customized process views," *Data & Knowledge Engineering*, vol. 64, pp. 419-438, 2008.
- [18] M. Shen and D. R. Liu, "Discovering role-relevant process-views for disseminating process knowledge," *Expert Systems with Applications*, vol. 26, pp. 301-310, 2004.
- [19] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security*, vol. 4, pp. 224-274, 2001.
- [20] E. Y. Li, T. C. Du, and J. W. Wong, "Access control in collaborative commerce," *Decision Support Systems*, vol. 43, pp. 675-685, 2007.