Knowledge-Based Systems 31 (2012) 41–54

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Modeling the knowledge-flow view for collaborative knowledge support

Duen-Ren Liu *, Chih-Wei Lin

Institute of Information Management, National Chiao Tung University, No. 1001 Ta Hseuh Rd., Hsinchu 300, Taiwan

ARTICLE INFO

Article history:
Received 11 January 2011
Received in revised form 16 January 2012
Accepted 25 January 2012
Available online 3 February 2012

Keywords:
Knowledge flow
Knowledge-flow view
Collaborative knowledge support
Teamwork
Ontology

ABSTRACT

In knowledge-based organizations, workers need task-relevant knowledge and documents to support their task performance. A knowledge flow (KF) represents the flow of an individual's or group members' knowledge-needs and the referencing sequence of documents in the performance of tasks. Through knowledge flows, organizations can provide task-relevant knowledge to workers to fulfill their knowledge-needs. Nevertheless, in a collaborative environment, workers usually have different knowledge-needs in accordance with their individual task functions. Conventional KF models do not provide workers with the different views of a knowledge flow that they require to meet these knowledge-needs. Several researchers have investigated KF models but they did not address the concept of the knowledge-flow view (KFV).

This study proposes a theoretical model of the KFV using innovative methods. Basically, a KFV is a virtual knowledge flow derived from a base knowledge flow that abstracts knowledge concepts for individual workers based on their knowledge-needs. The KFV model in this study builds knowledge-flow views by abstracting knowledge nodes in a base knowledge flow to generate corresponding virtual knowledge nodes through an order-preserving approach and a knowledge concept generalization mechanism. The knowledge-flow views not only fulfill workers' different knowledge-needs but also facilitate knowledge support in teamwork.

1. Introduction

In a knowledge-based organization, knowledge workers need to acquire a variety of knowledge (information) about their tasks [11]. Therefore, many organizations have built knowledge support platforms to assist workers in meeting their knowledge-needs. These platforms help workers to identify and share knowledge in order to speed up organization innovation and improve employee productivity [9,20]. Studies on formulating knowledge-needs and streamlining knowledge provision are becoming more prevalent as the value of knowledge support keeps increasing [2,28,33,39,49,50].

The fast pace of technology evolution and the short cycle time for solving problems in current knowledge intensive environments has led to an emphasis on teamwork [12,17]. For example, R&D activities often consist of many knowledge-intensive tasks that must be completed within a limited time period. These tasks are usually conducted through cross-function collaboration. By integrating the expertise and perspectives of various individuals, teams can quickly respond to interdisciplinary problems and enhance decision quality, thus providing a holistic solution. However, due to their individual task functions, many team members have different knowledge-needs; as a result, they may expend considerable effort in seeking and synthesizing knowledge to obtain the required task-relevant knowledge [37,52]. Reducing this expenditure of effort is one of the main challenges of collaborative knowledge support.

By mapping knowledge flows, organizations can provide task-relevant knowledge to workers that help them fulfill their knowledge-needs quickly and effectively [22]. A knowledge flow (KF) represents the flow of an individual's or group members' knowledge-needs and the referencing sequence of codified knowledge in conducting organizational tasks. Knowledge flows are an emerging topic of investigation in the knowledge management research field, and several studies have built knowledge-flow models to illustrate knowledge sharing among knowledge workers [20,22,26,30,32,55–58]. For example, researchers in scientific fields who propose new ideas through content publishing form a knowledge flow in science [57]. The known ideas in one paper inspire new ideas for other researchers, and the relationships or links established generate a citation chain. Some studies have addressed knowledge sharing by defining the process whereby knowledge is transferred from one team member to another [55,56,58]. Other researchers have focused on discovering knowledge flows by analyzing workers' knowledge-needs; the results have contributed to knowledge sharing whereby the codified knowledge becomes

* Corresponding author. Tel.: +886 3 5131245; fax: +886 3 5723792.
E-mail addresses: dliu@iim.nctu.edu.tw (D.-R. Liu), jwlin.iim93g@nctu.edu.tw (C.-W. Lin).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2012.01.014

available for recommendations to workers [22]. The shortcoming of these studies, however, is that the conventional models they adopt provide the same knowledge support to all team members; in other words, they do not consider the individual knowledge-needs that arise in a collaborative environment.

This study extends the previous research by exploring how to enhance conventional knowledge-flow models to satisfy workers' different knowledge-needs. The challenges in a collaborative team environment are considerable and they pose many barriers to knowledge flows [24,38]. Two of these barriers are low effectiveness and poor communication. Team members with different task functions require different conceptual levels of knowledge in order to conduct their tasks and communicate with each other. For example, workers need specific knowledge to perform their tasks and general knowledge to communicate with other workers whose tasks differ from their own. By taking the different conceptual levels of knowledge into consideration when identifying individual knowledge-needs, this work proposes a knowledge-flow view (KFV) model that aims to generalize knowledge concepts and derive knowledge-flow views; as such, the model would be capable of serving multiple knowledge-needs. A knowledge-flow view is a virtual knowledge flow derived from a base knowledge flow, employed to abstract knowledge concepts. The novel KFV model in this study uses an order-preserving approach and a knowledge concept generalization mechanism to abstract some knowledge nodes in a base knowledge flow, thus generating virtual knowledge nodes that correspond to the individual knowledge-needs of different workers. The proposed KFV model improves the knowledge support in collaborative teamwork environments and contributes to the literature on knowledge flow.

The rest of this paper is organized as follows: Section 2 provides a brief summary of related research. Section 3 builds a formal knowledge-flow model. Section 4 defines and analyzes a knowledge-flow view model and the algorithms to generalize knowledge concepts and derive knowledge-flow views. In Section 5, a mobile phone development process is exploited to demonstrate the knowledge-flow view application. Finally, Section 6 draws conclusions.

2. Related work

Knowledge is one of the key assets to ensure sustained competitive advantage in the highly technological and global environment of modern organizations [16,25,40,46]. To achieve success in this environment, workers need to effectively apply knowledge to successfully conduct knowledge-intensive operations and management activities [8,28,56]. Knowledge management (KM) supplies the principles of creation, organization, transfer and application of the knowledge within organizations [20] and is recognized as a crucial practice for enabling organizations to survive in a knowledge economy era [51]. One purpose of KM is to support workers in fulfilling their knowledge-needs, by bridging the gap between workers' knowledge and the requirements of various tasks [2,45,49]. Studies have shown that precise and timely knowledge support is an important mechanism for increasing both productivity and work effectiveness [21,28].

In a task-based business environment, tasks are conducted in work processes and the effective provision of task-relevant knowledge and context information is crucial to increasing workers' productivity. To meet this provision, solutions which integrate information retrieval (IR) and workflow management systems (WfMS) have been developed. These solutions proactively deliver task-relevant knowledge according to the context of tasks [1,33]. For example, the KnowMore system derives task profiles from process definitions that facilitate knowledge provision [1]. The FlowWiki system was developed by a wiki-based approach for agilely managing workflows and effectively providing relevant information to participators [19]. In this way, process participants can obtain knowledge that pertains to task profiles and/or the execution context of the current process.

To fulfill workers' knowledge-needs, knowledge flows provide links among knowledge sources. Through knowledge flows, workers can effectively obtain knowledge from these sources to execute tasks [20]. Knowledge flows illustrate the sequence of knowledge-needs and/or the order of referring documents when workers perform tasks. Knowledge flows can facilitate knowledge sharing and reuse in both business and research environments. For example, Zhuge [56] illustrated a knowledge flow within the software development team of a distributed organization. Here, the knowledge flow carried and gathered knowledge from one team member to another for sequential knowledge sharing. Similar knowledge sharing can take place in a citation chain where knowledge is transferred among scientific researches. In this context, the citation chain of papers is a knowledge flow that disseminates knowledge among scientists and inspires new ideas [57].

Several knowledge-flow models have been built in recent researches. Luo et al. [30] modeled a Textual Knowledge Flow (TKF) from a semantic link network. The purpose of the TKF was to recommend proper browsing paths to users after evaluating their interests and inputs. Lai and Liu [22] constructed a time-ordering knowledge flow model to illustrate the sequence of workers' knowledge referencing behaviors. In this model, workers obtained proper knowledge to fulfill their knowledge-needs through the knowledge flows discovered in document access logs. Kim et al. [20] proposed a knowledge flow model using a process-oriented approach to capture, store and transfer knowledge. Zhang et al. [53] used Petri-Net to model a knowledge flow. In this model, a knowledge node was used to generate, learn, process, understand, synthesize and deliver knowledge based on four types of flow relations: creation, merging, replication and broadcasting. Zhao and Dai [54] integrated business processes and knowledge flows and divided knowledge flows into sequence, distribution, combination and self-reflection patterns based on RAD (role-activity-diagram). Finally, Anjewierden et al. [4] suggested that the referencing sequence in weblogs may be regarded as a knowledge flow and can be described as a sender-message-receiver model.

Both knowledge flow and knowledge-based planning prompt similar ideas about embedding knowledge while building models. Knowledge-based planning is a planning methodology used to identify a sequence of tasks executed by one or more agents under given initial conditions and resource constrains to achieve final goals [5]. The methodology involves knowledge acquisition, knowledge validation and knowledge maintenance of planning domains, and adopts appropriate knowledge-based planning tools to build planning models [5]. For example, R-Moreno et al. [36] successfully utilized a planning and scheduling system as well as a workflow modeling tool to plan a telephone installation workflow model. The workflow modeling tool was used to acquire relevant knowledge, such as initial conditions, resource constrains and final goals; then the planning and scheduling system was used to convert the knowledge into planning standard expressions. A knowledge-based planning system can also be employed to manage the result of planned tasks for the purpose of fulfilling other tasks' preconditions. Chow et al. [10], for example, proposed a strategic knowledge-based planning system (SKPS) that combined knowledge rules with mathematical models to formulate co-loading shipment plans. Through SKPS, shipment planners could acquire, validate and maintain knowledge of the shipment domain, and thus build a co-loading shipment planning model so that executors could utilize the knowledge in the model to perform tasks efficiently.

As the above examples demonstrate, knowledge-based planning focuses on building planning models for problem solving or task execution. Knowledge flow research contributes to the building of knowledge-flow models for corresponding task execution plans (or workflow processes) that support knowledge provision, sharing and transferring [22,56]. Knowledge flows can be either derived by mining workers' access logs [22] or specified by knowledge-flow modelers according to their experience in executing the corresponding workflow process [55,58]. Besides these two methods, knowledge-based planning tools can complement knowledge flow research by helping modelers build the appropriate knowledge flows that correspond to task execution plans.

Ontology is a widely accepted approach for capturing and representing knowledge possessed by an organization [34,36,43]. It is a conceptualization mechanism that defines knowledge concepts in a specific domain and constructs a hierarchical structure to describe their inter-relationships [14]. Ontology can promote a common understanding throughout a whole organization to facilitate knowledge storage, retrieval and synthesis [35]. For example, the common terminologies and knowledge concepts in an ontology can improve the problem-solving capability and efficiency within a supply chain [6]. Another example of ontology pertains to the knowledge concepts derived from Wikipedia articles and categories, which can be used to predict the contents of documents [44]. Weng and Chang [47] proposed a research document recommendation system which exploited ontology to construct user profiles, and utilized the profiles to illustrate researchers' interests. Afacan and Demirkan [3] developed an ontology-based universal design support system to support designers in the conceptual design phase; it adopts ontologies to process and represent required knowledge. As the above examples illustrate, ontology is a versatile paradigm that can be applied in many domains.

Recently, business process modeling has been rapidly applied to streamline business administration and to facilitate cooperation among enterprises. Business process modeling refers to the design, analysis and execution of business processes [15]. Its goals are to describe a set of activities that can be performed in sequence, and to allocate resources and arrange jobs optimally by analyzing the organizational and technical environments [48]. By employing appropriate modeling tools, business process modeling can provide pre-defined templates that allow enterprises to enact their business processes in an effective and efficient manner.

In an industrial environment, processes describe the flows of business operations. Workflow management systems are definition and execution tools that support these operations [35]. In practice, participants involved in a workflow need a flexible workflow model capable of providing appropriate process information [2,27]. Because of the increasing complexity of business processes and the variety of participants, it is beneficial for organizations to define virtual processes with different views of the workflow [7,13,27,42]. Liu and Shen [27] presented a novel concept of process abstraction: the process-view. A process-view is an abstracted process derived from a base process to provide generalized process information. The process-view is generated by an order-preserving approach, which ensures that the original order of the activities in the base process is preserved. This paper adopts similar ideas to generate knowledge-flow views from a base knowledge flow, while retaining the knowledge referencing order.

3. Knowledge-flow model: a base knowledge flow

In cooperative working environments, a knowledge flow (KF) represents the flow of team members' knowledge-needs and the referencing sequence of codified knowledge that workers need while conducting business processes or research tasks. For example, a mobile phone development process consists of multiple tasks which require joint efforts from Marketing, Design, Outsourcing, Quality Assurance and Sales departments. Participants not only contribute their expertise, but also refer to additional codified knowledge that contributes to the performance of tasks in processes. The flow of knowledge-needs and the sequence of document reference can be represented by a knowledge flow. Fig. 1 shows a mobile phone development process consisting of nine tasks: business analysis, industrial design, major parts identification, parts sourcing, hardware design, platform setup, application design, verification and commercialization.

In the above process, team members may have the knowledge-needs of *Marketing segmentation* and *Consumer analysis* while conducting the Business analysis task. The knowledge concepts relevant to the knowledge-needs include: *Geographic segmentation*, *Psychographic segmentation*, *Consumption environment* and *Consumer behavior*. Knowledge-flow modelers put these knowledge concepts into a knowledge node to represent the knowledge-needs of the Business analysis task. In addition, the team members may also have the knowledge-needs of accessing *Compliance guidance* and *Usability checklist*, two related knowledge concepts, while performing the Verification task [18]. The knowledge of how to build knowledge flows is derived from structured interviews and workshops [20], system event logs [22], as well as the content of tasks. For example, in investigating the whole business process, knowledge-flow modelers rely on their experience [55,58], interviews of domain experts [20], and/or the analyses of workers' document access logs [22,26] to collect knowledge-needs on a task-by-task basis. These knowledge-needs are illustrated by knowledge concepts which are identified by domain ontology. By using domain ontology, the knowledge-flow modelers group relevant knowledge concepts into corresponding knowledge nodes to form a knowledge flow.

Fig. 2 shows the corresponding knowledge flow of the mobile phone development process. In the knowledge flow, for example, the knowledge concept *Consumer behavior* is related to market trends research and customer preferences investigation, which facilitate marketing staff and designers in identifying major parts such as display, battery and cards options by evaluating their combinations. Accordingly, the knowledge-flow modelers group the relevant knowledge concepts *Consumer behavior*, *Display options*, *Battery options* and *Card options* to form the knowledge node k_2 to represent the knowledge-needs of conducting the major parts identification task.

A knowledge flow that represents knowledge-needs and a reference sequence is herein termed a *base knowledge flow*. In this section, we formally define domain ontology and base knowledge flow for the purpose of building a theoretical knowledge-flow model. Definition 1 models domain ontology, which is the infrastructure for sharing knowledge concepts throughout the whole organization. Definitions 2–7 formulate the knowledge-flow model.

Definition 1 (*Domain ontology*). Ontology is constructed to define knowledge concepts and their hierarchical relationships in a domain.

Ontology is defined as $O = \langle C, HR \rangle$, where C is a set of knowledge concepts derived from a specific domain. HR is a set of hierarchical relations which define the parent-child relationships among knowledge concepts in C, and HR is formally expressed by $HR = \{hr | hr \in C \times C\}$.

For two knowledge concepts, x and y, if x has a downward link to y (or y has an upward link to x) in an ontology, 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 relative conceptual level of two knowledge concepts. Relations between the parent concept x and the child concept y are formally ex-

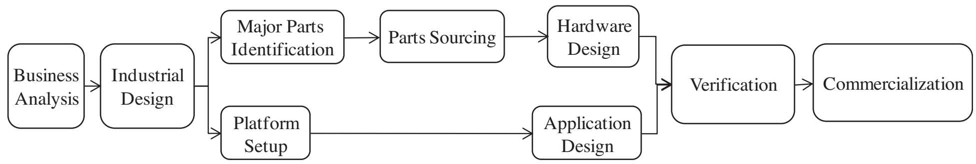

Fig. 1. Mobile phone development process.

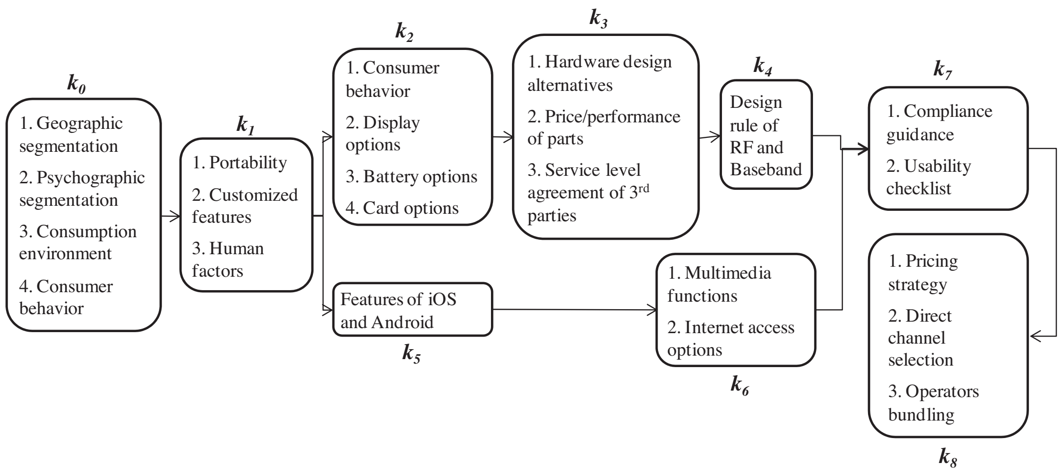

Fig. 2. Knowledge flow of the mobile phone development process.

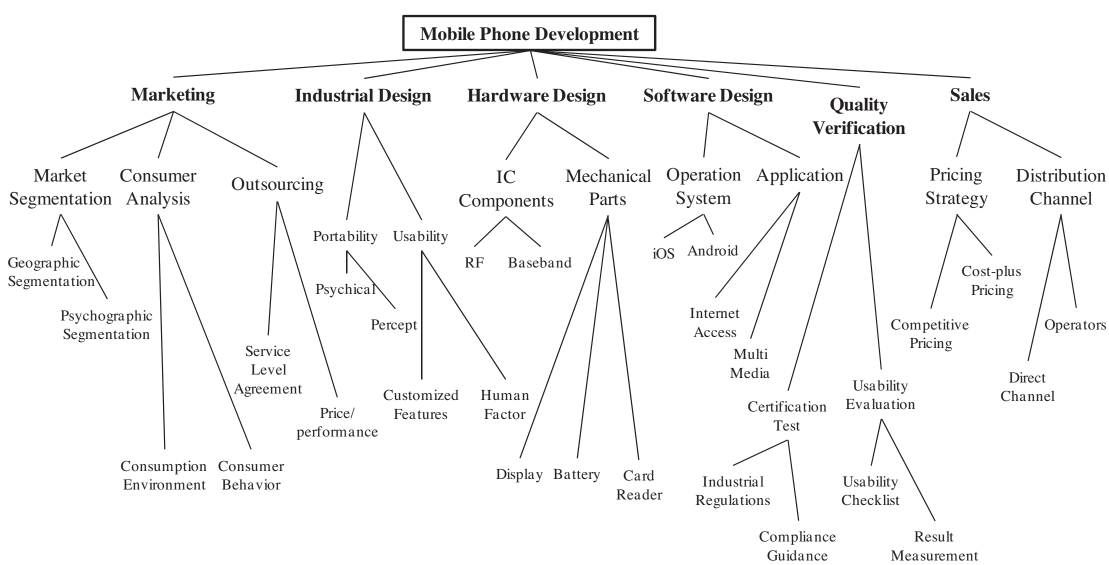

Fig. 3. Domain ontology of mobile phone development.

pressed by *Specialization* $(x) = \{y|y$ is a child concept of $x\}$ and *Generalization*$(y) = \{x|x$ is a parent concept of $y\}$.

Fig. 3 shows an example of ontology in the mobile phone development domain. The root of the ontology is *Mobile Phone Development*. It represents the most general knowledge concept, as indicated also by R&D strategy and the product development guideline. Six subconcepts, *Marketing*, *Industrial design*, *Hardware design*, *Software design*, *Quality verification* and *Sales* appear under *Mobile Phone Development*. Likewise, *Market segmentation*, *Consumer analysis* and *Outsourcing* are the subconcepts of *Marketing*. Hence, *Specialization* (*Marketing*) = {*Market segmentation*, *Consumer analysis*, *Outsourcing*} and *Generalization* (*Market segmentation*) = {*Marketing*}.

In existing research [23,29,31,41,43], the relations of ontologies are designated as: *is-a*, *part-of*, *subclass*, *synonym* or *related-to*. The meanings of these relations pertain to the design purpose of ontologies and the characteristics of knowledge concepts. In this work, ontology is designed to represent knowledge concepts and their hierarchical relationships in a domain. The use of ontology facilitates the abstraction of knowledge concepts based on their conceptual levels, which are required for building knowledge-flow views. Two semantic relations, *Generalization* and *Specialization*, are used to describe the relative conceptual levels of two knowledge concepts without distinguishing between the meanings of relations such as *is-a*, *subclass*, *synonym* or *related-to*. According to the *Generalization* relation, child (specific) knowledge concepts can be abstracted to parent (general) knowledge concepts. For example, the knowledge concept *Operation system* contains knowledge about categories and functions of APIs. Knowledge concept *iOS* contains knowledge of detailed specifications of Apple iOS's APIs. Thus, these two knowledge concepts are related and on different conceptual levels. The knowledge concept *Operation system* comprises more general knowledge than does knowledge concept *iOS*. Thus, knowledge concept *Operation system* is on a higher conceptual level than knowledge concept *iOS*. The same relation exists between knowledge concepts *Operation system* and *Android*. As shown in Fig. 3, knowledge concepts *iOS* and *Android* have upward links to knowledge concept *Operation system*. The relations among them can be expressed as *Generalization* (*iOS*) = {*Operation system*} and *Specialization* (*Operation system*) = {*iOS*, *Android* ...}.

Based on domain ontology, knowledge-flow modelers can formulate workers' knowledge-needs by using combinations of knowledge concepts on different conceptual levels. For example, workers with knowledge-needs about market segmentation might identify and divide potential consumers into groups according to their characteristics, behavior and location. In this example, knowledge-needs can be represented either by the knowledge concept *Market segmentation* or by two knowledge concepts, *Geographic segmentation* and *Psychographic segmentation*. The knowledge concept *Market segmentation* is a general concept. As such, it is used to describe the purpose of segmentation or to introduce the guideline of selecting one segmentation alternative among others. By contrast, the knowledge concepts, *Geographic segmentation* and *Psychographic segmentation*, are specific concepts that describe detailed knowledge pertaining to the steps used in analysis or the segmentation criteria. *Market segmentation* is the parent (general) concept of *Geographic segmentation* and *Psychographic segmentation*, whereas *Geographic segmentation* and *Psychographic segmentation* are the child (specific) concepts of *Market segmentation*.

As this example shows, workers' knowledge-needs can be expressed as a combination of knowledge concepts in domain ontology, where the conceptual levels of these knowledge concepts can be identified by their positions in the ontology. By grouping knowledge concepts at the proper conceptual levels, knowledge-flow modelers can use domain ontology as a reference base to identify workers' knowledge-needs. Furthermore, domain ontology can facilitate the abstraction of knowledge concepts, which are required for generating knowledge-flow views.

Next, we formulate the knowledge-flow model through a series of definitions provided below:

Definition 2 (*Knowledge node*). A knowledge node x is a set of knowledge concepts needed by workers to fulfill their tasks. The knowledge concepts of x are denoted as $KC(x) = \{c_1, c_2, c_3, \ldots, c_m\}$ where knowledge concept c_i can be identified by domain ontology.

Definition 3 (*Dependency*). A dependency is an ordered pair (knowledge node x, knowledge node y) denoted by $dep(x, y)$. This notation indicates that after knowledge concepts in x have been referenced, workers can start to reference the knowledge concepts in y. In $dep(x, y)$, x is called the preceding node and y is called the succeeding node.

Definition 4 (*Knowledge flow*). A knowledge flow (KF) is a 2-tuples $\langle KNS, DS\rangle$, where KNS is a nonempty set, and its members are knowledge nodes in the knowledge flow. DS is a nonempty set, and its members are dependencies that preceding nodes and succeeding nodes are in KNS.

Definition 5 (*Neighboring*). Two knowledge nodes are neighboring if a dependency between them exists in DS.

Definition 6 (*Path*). Given a knowledge flow $KF = \langle KNS, DS\rangle$, a path is defined to include a starting knowledge node k_0, intermediate knowledge nodes $k_1, k_2, \ldots, k_{n-1}$, an ending knowledge node k_n and a set of dependencies, $dep(k_{i-1}, k_i) \in DS$, for $i = 1, 2, \ldots, n$. The path from k_0 to k_n is denoted by $k_0 \rightarrow k_n$.

Definition 7 (*Ordering relation*). Given a knowledge flow $KF = \langle KNS, DS\rangle$ and two knowledge nodes $x, y \in KNS$, knowledge node x has a higher order than knowledge node y if a path $x \rightarrow y$ exists. The ordering relation is denoted as $x > y$.

4. Knowledge-flow view model: a virtual knowledge flow

By knowing what other members know, a team is able to gain better decision quality and communicate more effectively [37]. Therefore, team members not only need specific knowledge to conduct the tasks assigned to them, but also require general knowledge about tasks performed by other members to facilitate their communication. For example, in the mobile phone development process, marketing staff members refer to specific geographic segmentation documents to identify possible consumer groups, and they gather specific knowledge of consumer behavior to determine the acceptance level of a new mobile phone. In addition, they need general knowledge related to *Industrial design*, *Hardware design*, *Software design*, *Quality verification* and *Sales* to communicate with members outside their departments through the use of common terminology. The knowledge support of both specific and general knowledge pertaining to different tasks can assist marketing staff members to complete their business analysis task and increase the communication quality of the team. However, since conventional knowledge-flow models provide only a single view of a knowledge flow and do not consider personalized requirements, they are not applicable in such environments. In fact, project managers do not need specific and detailed knowledge about Business analysis, Industrial design, Hardware design and other tasks. They only need general knowledge of these tasks to help them make decisions and communicate with other team members. Fig. 4 shows a knowledge-flow view with general knowledge concepts that can meet project managers' knowledge-needs.

The knowledge-flow view in Fig. 4 includes three virtual knowledge nodes: vk_1, vk_2 and vk_3, which represent the knowledge-needs of project managers in the mobile phone development process. The virtual knowledge node vk_1 consists of two general knowledge concepts: *Market segmentation* and *Consumer analysis*, which project managers require to oversee the business analysis task. These two general knowledge concepts are abstracted from four specific knowledge concepts: *Geographic segmentation*, *Psychographic segmentation*, *Consumption environment* and *Consumer behavior*. In node vk_2, which represents product design-related knowledge concepts at the general conceptual level, the three general concepts: *Industrial design overview*, *Hardware design overview* and *Software design overview* are more helpful to project managers in communicating with product designers than the concepts from which they are abstracted. Finally, the general knowledge concepts in node vk_3 are advantageous to project managers in overseeing Verification and Commercialization tasks; hence, the knowledge-flow view in Fig. 4 appropriately formulates the knowledge-needs of project managers in the development process, and illustrates corresponding knowledge concepts at the proper conceptual levels.

The formal model of the knowledge-flow view (KFV) can now be defined: Knowledge-flow views are the abstracted forms of a base knowledge flow, and are herein referred to as *virtual knowledge flows*. Since knowledge-flow views are abstractions, different

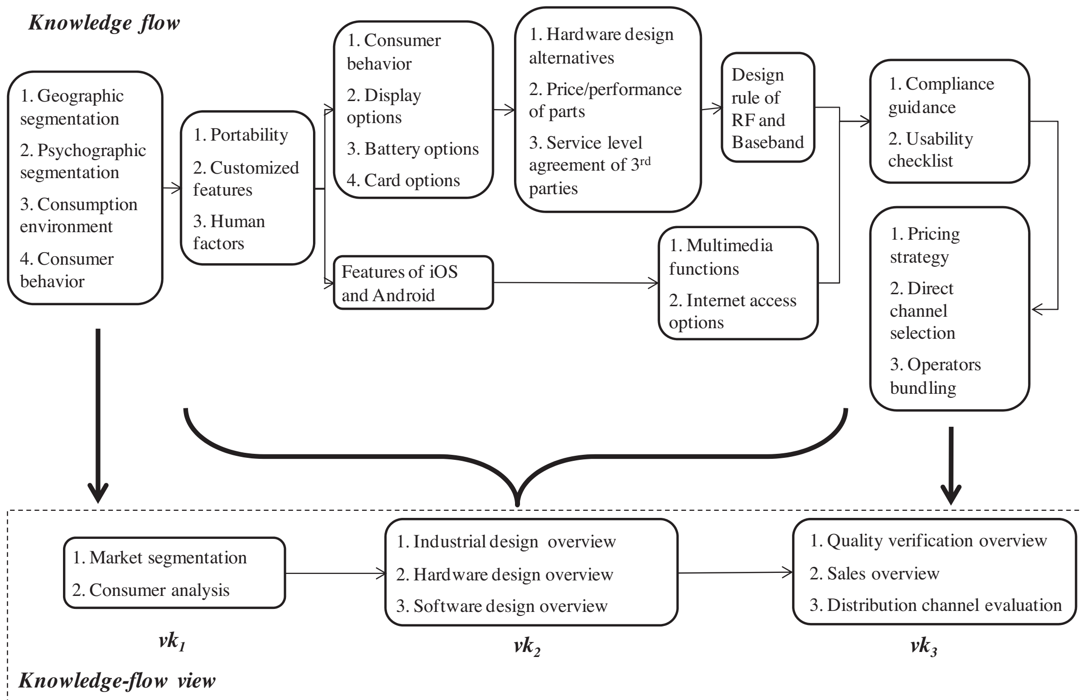

Fig. 4. A knowledge-flow view for project managers.

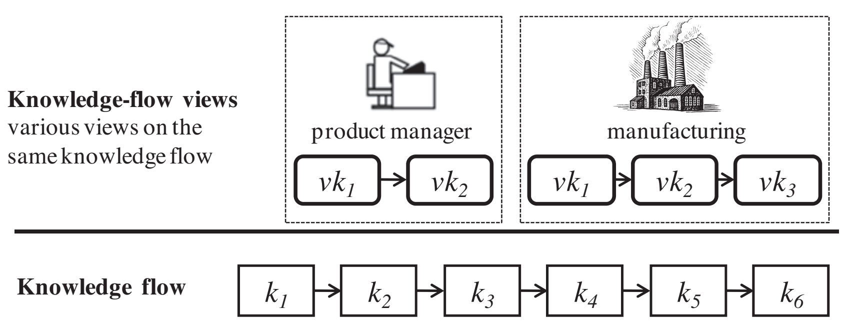

Fig. 5. Illustrative examples of knowledge-flow views.

knowledge-flow views can be generated based on individual participants' knowledge-needs and organization policies. By providing different knowledge-flow views that hide all or some of the detailed information in a base knowledge flow, organizations can be better equipped to enforce policies. Fig. 5 shows an example of mapping a base knowledge flow to multiple knowledge-flow views. While product managers do not need to have detailed knowledge of all the knowledge concepts in the knowledge flow, they must have general marketing knowledge to understand marketing trends and to increase communication effectiveness within a team. To serve product managers' knowledge-needs, knowledge-flow modelers can abstract marketing-related knowledge nodes and generalize knowledge concepts in those nodes to hide detailed marketing information. A possible knowledge-flow view for product managers is as follows: knowledge nodes k_1 and k_2 are abstracted to a virtual knowledge node vk_1, and k_3, k_4, k_5 and k_6 are abstracted to vk_2. In addition, manufacturers have their own knowledge flow view which contains specific manufacturing knowledge (represented by vk_2), general knowledge of marketing and design (represented by vk_1), as well as general knowledge of sales and post service (represented by vk_3). As this illustrative analysis shows, a knowledge flow can be abstracted to multiple knowledge-flow views by considering different knowledge-needs and organization policies. In this way, workers can obtain the proper knowledge-flow views that help them acquire the knowledge support they need in collaborative environments.

4.1. The formal framework of the knowledge-flow view model

Definitions 8–13 describe the properties and basic terms that constitute the theoretical framework of the knowledge-flow view model.

Definition 8 (*Concealing criteria*). A concealing criterion is a 3-tuples ⟨worker *w*, knowledge node *kn*, boolean of abstraction *Y*/*N*⟩, which states whether the knowledge concepts of a knowledge node *kn* are too specific or confidential for a worker *w*'s task functions. If the answer is yes, the boolean of abstraction is set to *Y*, and the knowledge concepts of *kn* are abstracted. On the other hand, if the knowledge concepts of *kn* are appropriate for *w*'s task functions and need not be abstracted, the boolean of abstraction is set to *N*.

The concealing criteria are defined by knowledge-flow modelers to comply with the company's information security control rules and fulfill team members' need-to-know requirements. The knowledge-flow modelers can refer to their experience or utilize experts' knowledge to discover what knowledge nodes should be abstracted if the knowledge concepts in the knowledge nodes are too specific or confidential for workers to perform their task functions properly.

Two scenarios illustrate how the knowledge-flow modelers define concealing criteria when deriving VKFs from a base KF for a sourcing planner (denoted as p for short). The sourcing planner oversees the management of outsourced parts, including: surveying reliable suppliers, evaluating price/performance of parts and negotiating service level agreements with suppliers. The base KF (shown in Fig. 2) includes two knowledge nodes, k_2 and k_4, which contain the required knowledge concepts to conduct two tasks: major parts identification task and hardware design task, respectively. In the first scenario, the concealing criterion is made by the knowledge-flow modelers, based on information security control rules. In order to sustain their competitive advantage, many companies enforce information security policies to protect their precious intellectual properties, such as hardware design specifications, from unauthorized access. Only work-related employees can access such specific knowledge or information. The sourcing planner's tasks are not directly related to hardware design tasks, so p is not allowed to access the knowledge concept Design rule of RF and Baseband which is one type of hardware design specification. Because the knowledge concept Design rule of RF and Baseband is included in k_4 and p is not allowed to access it, the knowledge-flow modelers define a concealing criterion $\langle p, k_4, Y\rangle$ while deriving VKFs for p. The concealing criterion $\langle p, k_4, Y\rangle$ indicates that k_4's knowledge concept Design rule of RF and Baseband needs to be abstracted to a general knowledge concept, IC components, based on the domain ontology (as shown in Fig. 3). The concealing criterion not only protects the specific knowledge concept Design rule of RF and Baseband from unauthorized access, but also provides general knowledge concept IC components for p to effectively communicate with other team members.

In another scenario, the concealing criterion is made in terms of workers' need-to-know requirements. Supposing that the knowledge concepts *consumer behavior*, *display options*, *battery options* and *card options* in k_2 are too specific for p to conduct his/her tasks. Consequently, the knowledge-flow modelers define a concealing criterion $\langle p, k_2, Y\rangle$ to reflect p's knowledge-needs when deriving VKFs for p. The concealing criterion $\langle p, k_2, Y\rangle$ indicates that k_2's knowledge concepts need to be abstracted to the general knowledge concepts *consumer analysis* and *mechanical parts*, respectively, based on the domain ontology (as shown in Fig. 3). The two scenarios show that the knowledge for defining concealing criteria is practical and context-dependent, depending on the consideration of security as well as the knowledge-needs of the participants.

Definition 9 (*Virtual knowledge node*). A virtual knowledge node consists of a set of knowledge nodes and corresponding knowledge concepts. The knowledge concepts of a virtual knowledge node are abstracted from the knowledge concepts of the corresponding knowledge nodes. A virtual knowledge node vx is a 2-tuples $\langle ANS, AKC\rangle$, where *ANS* (Abstracted Knowledge Node Set) is a nonempty set and its members are knowledge nodes or previously defined virtual knowledge nodes. *AKC* (Abstracted Knowledge Concept Set) is a nonempty set and its members are knowledge concepts defined in domain ontology.

The knowledge concepts of vx are denoted as $AKC(vx) = \{c_1, c_2, c_3, ..., c_q\}$ where knowledge concept c_i can be identified in domain ontology.

Definition 10 (*Virtual dependency*). Given $KF = \langle KNS, DS\rangle$ and two virtual knowledge nodes, vx and vy, a virtual dependency $vdep(vx, vy)$ from vx to vy exists if $dep(x, y)$ is in DS, where x is a member of vx and y is a member of vy. A virtual dependency is used to connect two virtual knowledge nodes, vx and vy.

Definition 11 (*Virtual knowledge flow*). A virtual knowledge flow (*VKF*) is a 2-tuples $\langle VKNS, VDS\rangle$, where *VKNS* is a nonempty set and its members are virtual knowledge nodes, and *VDS* is a nonempty set and its members are virtual dependencies. A knowledge-flow view is herein termed a virtual knowledge flow.

Definition 12 (*Virtual Path*). Given a virtual knowledge flow $VKF = \langle VKNS, VDS\rangle$, a *virtual path* in *VKF*, extending from vk_0 to vk_n, is a sequence of virtual knowledge nodes $vk_0, vk_1, vk_2, ..., vk_n \in VKNS$, such that $vdep(vk_{i-1}, vk_i) \in VDS$ for $i = 1, 2, \ldots, n$. The virtual path from vk_0 to vk_n is denoted as $vk_0 \rightarrow vk_n$.

Definition 13 (*Virtual ordering relation*). Given a virtual knowledge flow $VKF = \langle VKNS, VDS\rangle$ and two virtual knowledge nodes: vx and $vy \in VKNS$, vx has a higher order than vy if a virtual path $vx \rightarrow vy$ exists. The virtual ordering relation is denoted as $vx > vy$.

Fig. 6 illustrates the relationship between the components of the novel model. As the figure shows, a knowledge-flow view is an abstraction from a base knowledge flow. The abstraction relationships exist in major components. Virtual knowledge nodes are abstracted from knowledge nodes; thus, a virtual knowledge node contains generalized knowledge concepts that are abstracted from the knowledge concepts in corresponding knowledge nodes. Both the abstracted knowledge concepts and the concepts from which they are abstracted can be identified in the domain ontology.

4.2. An order-preserving approach for deriving a knowledge-flow view

Liu and Shen [27] presented an order-preserving approach to the generation of virtual processes from a base process in workflow environments. The approach is designed to ensure that the original ordering relation of activities in a base process is preserved in virtual processes. This paper adopts the order-preserving approach for the purpose of generating knowledge-flow views from a base knowledge flow, that retain their knowledge referencing order in the base knowledge flow. A legal virtual knowledge node must follow three basic rules to preserve the ordering property in a virtual knowledge flow. The basic rules are membership, atomicity and ordering preservation.

Rule 1 (*Membership*). A virtual knowledge node may be abstracted from either knowledge nodes or previously defined virtual knowledge nodes. The membership among knowledge nodes and virtual knowledge nodes is transitive. If x is a member of y and y is a member of z, then x is a member of z.

Rule 2 (*Atomicity*). A virtual knowledge node is an atomic unit of knowledge access. A virtual knowledge node is activated for knowledge access if, and only if, one of its members is activated to refer knowledge. On the other hand, a virtual knowledge node has completed its knowledge access if, and only if, all of its members have completed their knowledge access.

Moreover, if an ordering relation (>) between two virtual knowledge nodes exists in a virtual knowledge flow, the implied ordering relation between the respective members of the two virtual knowledge nodes is ">" due to the atomicity rule.

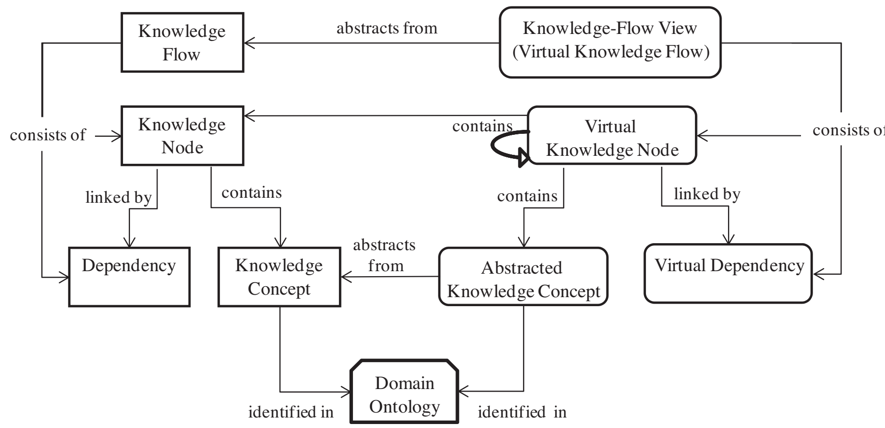

Fig. 6. Knowledge-flow view model.

Rule 3 (*Ordering preservation*). The implied ordering relation between two virtual knowledge nodes' respective members must conform to the ordering relation in the base knowledge flow.

The procedure for deriving knowledge-flow views from a base knowledge flow is described as follows: Knowledge-flow modelers select some essential knowledge nodes based on team members' knowledge-needs and/or the company's information security control rules to conceal detailed information.

Definition 14 (*Essential knowledge node*). An essential knowledge node is a knowledge node selected by knowledge-flow modelers for the purpose of generating a virtual knowledge node and generalizing knowledge concepts. To conceal confidential or detailed information, one or more knowledge nodes in a base knowledge flow should be selected as the essential knowledge node(s).

There are three sets of knowledge nodes: (a) The Essential Knowledge Node Set (*ENS*) represents the knowledge nodes selected by knowledge-flow modelers; (b) The Expanding Knowledge Node Set (*ES*) includes the knowledge nodes in *ENS* and the knowledge nodes which are added due to order-preserving property; and (c) The Neighboring Knowledge Node Set (*NNS*) represents the neighboring (adjacent) knowledge nodes to the knowledge nodes in *ES*. The knowledge nodes in *NNS* are candidates to be added to the *ES* for preserving the ordering property of a virtual knowledge node. If the implied ordering relation between any knowledge node in *NNS* and any knowledge node in *ES* does not comply with the original ordering relation in the base knowledge flow, the violated knowledge nodes in *NNS* should be incorporated into *ES*. Definition 15 defines a minimum expanding knowledge node set (*MES*) to ensure that only the necessary knowledge nodes are added, thus preserving the ordering relation while expanding *ES*.

Definition 15 (*Minimum expanding knowledge node set, MES*). This set includes both the essential knowledge nodes and the minimum required knowledge nodes which are added to preserve the ordering relation in virtual knowledge nodes. The implied ordering relation between any knowledge node in *ES* and any knowledge node not in *ES* must comply with the original ordering relation in the base knowledge flow. Note that an ES_i (a superset of *ENS*) is a *MES* if ES_i satisfies the order-preserving property, and the ES_i does not contain other ES_j (a superset *ENS*) that also satisfies the order-preserving property. The *MES* only contains the essential knowledge nodes and the required knowledge nodes to preserve ordering relations. Based on the *MES*, one can generate virtual knowledge nodes and virtual dependencies, as well as derive knowledge concepts of virtual knowledge nodes.

4.3. The procedure for discovering the minimum expanding knowledge node set

For a given knowledge flow, $KF = \langle KNS, DS \rangle$, and an essential knowledge node set, *ENS*, Fig. 7 shows the procedure for discovering the minimum expanding knowledge node set *MES*. Initially, the algorithm creates a working set ES_1 of the expanding knowledge node set (*ES*) that initially equals to the essential knowledge node set (*ENS*). According to the ordering preservation rule and *ES* definition, $\forall x \in KNS$, $x \notin ES$, the implied ordering relation between x and all members of *ES* must conform to the ordering relations in the knowledge flow, *KF*. *ENS* is the starting point to the discovery of *MES*. A while loop (lines 7–10) repeatedly finds any neighboring knowledge node that violates the ordering relation. If a neighboring knowledge node violates the ordering relation conditions (line 9), it is added into *ES*. Finally, $\forall x \in KNS$ and $x \notin ES$; if the implied ordering relations between x and all of the members of *ES* satisfy the ordering preservation, the repeat-until loop stops at this point. The final *ES* is the *MES*, which is the knowledge node set *ANS* of a virtual knowledge node, *vx*, derived from *ENS*.

4.4. The procedure for discovering virtual dependencies

All virtual knowledge nodes can be derived from a knowledge flow $KF = \langle KNS, DS \rangle$ that form the *VKNS* of a virtual knowledge flow *VKF*, by repeatedly executing the procedure in Fig. 7. For any pair of *VKNS*'s members, *vx* and *vy*, the virtual dependency *vdep*(*vx*, *vy*) exists if *dep*(x, y) exists in *DS*, where x is a member of *vx* and y is a member of *vy*.

4.5. The procedure for deriving knowledge concepts of a virtual knowledge node

After a virtual knowledge node, *vx*, has been derived, the knowledge concepts of *vx* should be derived. Fig. 8 shows the procedure for deriving the knowledge concepts of a virtual knowledge node. Let *ECS* (Essential Concept Set) denote the set of knowledge concepts of essential knowledge nodes that need to be concealed or hidden. A *minimum generalization policy* is used to generalize (conceal) the concepts in *ECS*. For each concept *c* in *ECS*, the parent concept of *c* in the ontology is selected to form the knowledge concept set (abstracted knowledge concept set, *AKC*) of the virtual knowledge node, *vx*. On the other hand, for some knowledge nodes that are in *MES* but not in *ENS*, the corresponding knowledge concepts do not need to be generalized and are directly included in *AKC* of *vx*. Initially, *AKC* is derived from the generalization of the knowledge concepts in *ECS*. Then, *AKC* incorporates the knowledge

```
(1) procedure DiscoverMES (INPUT: knowledge flow KF=<KNS, DS>, essential knowledge node set ENS,
                          OUTPUT: virtual knowledge node vx = <ANS, AKC>)
(2) begin
(3)   expanding knowledge node set ES = ENS
(4)   repeat
(5)     Working Set 1 ES_1 = ES
(6)     Neighboring Knowledge Node Set NNS = {x|∀x, y ∈ KNS, x ∉ ES, y ∈ ES and dep (x, y) ∈ DS}
(7)     while NNS is not empty do
(8)       Select a knowledge node x from NNS and remove x from NNS
(9)       if (∃ y, z ∈ ES, such that x > y exists in KF and x > z does not exist in KF) or
             (∃ y, z ∈ ES, such that y > x exists in KF and z > x does not exist in KF) then Add x into ES
(10)    end while
(11)  until ES_1 = ES
(12)  MES = ES
(13)  ANS of vx = MES
(14) end
```

Fig. 7. Procedure for discovering the minimum expanding knowledge node set, *MES*.

concepts of the knowledge nodes in *MES* but not in *ENS*. If some knowledge concepts in *AKC* are members of *ECS* or the descendant concepts of *ECS*'s members, they are removed from *AKC* due to the concept concealing requirement. The final step is to remove the implied (redundant) concepts from *AKC*; hence, the knowledge concepts of *vx* can be obtained.

If knowledge-flow modelers want to add or delete knowledge concepts in a KFV, it is appropriate to do these operations in the corresponding base knowledge flow and then re-generate a new KFV to replace the old one. A KFV is derived from a base knowledge flow. Conceptually, it is difficult to map back any changes in a KFV to the corresponding base knowledge flow. Thus, these operations should be made in the base knowledge flow. A similar concept exists in database management systems, where administrators modify the definition of a database view by adding or deleting fields in the underlying base tables.

Definition 16 (*Implied concept*). A concept *c* is implied under a concept set *C* if *c* can be inferred by other concepts in *C*. Based on a domain ontology, the concept *c* is mapped to an ontology node *k* that has *n* child ontology nodes k_i ($i = 1, \ldots, n$). The concept *c* is an implied concept if each k_i's corresponding concept is either in concept set *C* or can be implied by other concepts in concept set *C*.

5. Case illustration and analysis

This section uses a knowledge flow of a mobile phone company, named Smart-Tech Company, to illustrate the application of the knowledge-flow view. The knowledge flow represents the knowledge-needs that a project team requires when conducting a mobile phone development process in the company. According to the process, knowledge-flow modelers consult domain experts and team participants to acquire important knowledge-needs and identify corresponding knowledge concepts for the purpose of representing knowledge-needs in a knowledge flow.

In this company, the mobile phone development team requires participants from various departments. Those team members have different task functions: (1) the project manager controls and coordinates the project, (2) the marketing analyzer conducts the business analysis, (3) the designer is responsible for product design, (4) the salesperson focuses on product commercialization, (5) the inspector carries out the quality assurance tasks, and (6) the sourcing planner oversees the management of outsourced parts. Based on different knowledge-needs, knowledge-flow modelers can design knowledge-flow views for individual participants.

The following discussion pertains to the sourcing planners at this company whose task function is parts outsourcing. First, knowledge-flow modelers make the concealing criteria for the sourcing planners, as required by the information security policy of the company and in consideration of the information granularity suggested by domain experts. Hence, the *Essential Knowledge Nodes* are identified based on the concealing criteria and all knowledge concepts in the *Essential Knowledge Nodes* should be included in an *Essential Concept Set*. Then, a virtual knowledge node is obtained by the order-preserving approach to ensure that the ordering in the base knowledge flow is retained. Finally, the knowledge-flow modelers abstract the knowledge concepts in the *Essential Concept Set* using the domain ontology and the minimum generalization policy.

The knowledge flow in Fig. 9 includes nine knowledge nodes, k_0 to k_8, where each knowledge node contains multiple knowledge concepts. The knowledge concepts of k_2 are too specific for the sourcing planners, so the knowledge-flow modelers make a concealing criterion $\langle$*sourcing planner*, k_2, *Y*$\rangle$ to meet their knowledge-needs. Another concealing criterion $\langle$*sourcing planner*, k_4, *Y*$\rangle$ is also made because the knowledge concepts of k_4 are confidential for the sourcing planners. Following the two concealing criteria,

```
(1) procedure DeriveKnowledgeConcepts (INPUT: minimum expanding knowledge node set MES,
                                               essential knowledge node set ENS,
                                       OUTPUT: virtual knowledge node vx = <ANS, AKC>)
(2) begin
(3)   Essential Concept Set ECS = empty set
(4)   AKC(vx) = empty set
(5)   for each knowledge node x in ENS do ECS = ECS ∪ KC (x)
(6)   for each knowledge concept c in ECS do AKC(vx) = AKC(vx) ∪ {Generalization (c)}
(7)   for each knowledge node y in MES but not in ENS do AKC(vx) = AKC(vx) ∪ KC(y)
(8)   for each knowledge concept c in AKC(vx) do
        if c is in ECS or c is a descendant concept of ECS's member then remove c from AKC(vx)
(9)   for each knowledge concept c in AKC(vx) do
        if c is an implied concept under AKC(vx) then remove c from AKC(vx)
(10)  return AKC(vx)
(11) end
```

Fig. 8. Procedure for deriving knowledge concepts of a virtual knowledge node.

the knowledge-flow modeler selects two knowledge nodes, k_2 and k_4, as the essential knowledge nodes for the sourcing planners.

The knowledge-flow modeler applies the procedure in Fig. 7 to obtain a virtual knowledge node. Initially, the neighboring knowledge node set $NNS = \{k_1, k_3, k_7\}$ and ENS (essential knowledge node set) = ES (expanding knowledge node set) = $\{k_2, k_4\}$. Knowledge node k_3 is added into ES since the ordering of k_2 is higher than the ordering of k_3 (i.e. $k_2 > k_3$), but the ordering of k_4 is not higher than the ordering of k_3 (i.e. $k_4 \not> k_3$). Knowledge node k_7 is not added into ES since $k_2 > k_7$ and $k_4 > k_7$. Knowledge node k_1 is not added into ES since $k_1 > k_2$ and $k_1 > k_4$. Therefore, ES is changed to $\{k_2, k_3, k_4\}$. In the second execution, $NNS = \{k_1, k_7\}$ and $ES = \{k_2, k_3, k_4\}$. Knowledge node k_1 and knowledge node k_7 are not added into ES because the implied ordering relations between each member in NNS and ES satisfy the ordering preservation rule. Therefore, the execution stops. The minimum expanding knowledge node set MES includes knowledge nodes $\{k_2, k_3, k_4\}$, and a virtual knowledge node vk_1 is derived as shown in Fig. 10.

After discovering the minimum expanding knowledge node set MES, the knowledge-flow modeler uses the procedure in Fig. 8 and the ontology in Fig. 11 to derive the knowledge concepts of vk_1 based on the minimum generalization policy.

Initially, the essential concept set ECS equals to {*Consumer behavior*, *Display options*, *Battery options*, *Card options*, *Design rule of RF and Baseband*}. After performing the *Generalization* function on ECS, the knowledge-flow modeler obtains the abstracted knowledge concept set, AKC = {*Consumer analysis*, *Mechanical parts*, *IC components*}. Then, the knowledge concepts *Hardware design alternatives*, *Price/performance of parts* and *Service level agreement of third parties* are added in AKC, since knowledge node k_3 is in MES but not in ENS. Hence, AKC = {*Consumer analysis*, *Mechanical parts*, *IC components*, *Hardware design alternatives*, *Price/performance of parts*, *Service level agreement of third parties*}. The knowledge concepts of vk_1 can thus be obtained, as shown in Fig. 12.

Fig. 12 shows that vk_1 has a redundant knowledge concept, *Hardware design alternatives*, that can be inferred by *Mechanical parts* and *IC components*. Fig. 13 shows the result after removing the concept *Hardware design alternatives*. The final knowledge concepts of vk_1 are: *Consumer analysis*, *Mechanical parts*, *IC components*, *Price/performance of parts* and *Service level agreement of third parties*.

The example given demonstrates the knowledge-flow view from the sourcing planners' perspective. If the sourcing planners are not satisfied with the view that has been generated, knowledge-flow modelers can repeat the same steps to abstract a new knowledge-flow view by identifying other knowledge nodes as essential knowledge nodes. Similarly, it is possible to create other knowledge-flow views from other members' perspectives. Hence, the proposed knowledge-flow view model can enhance conventional knowledge flow models by supporting different team mem-

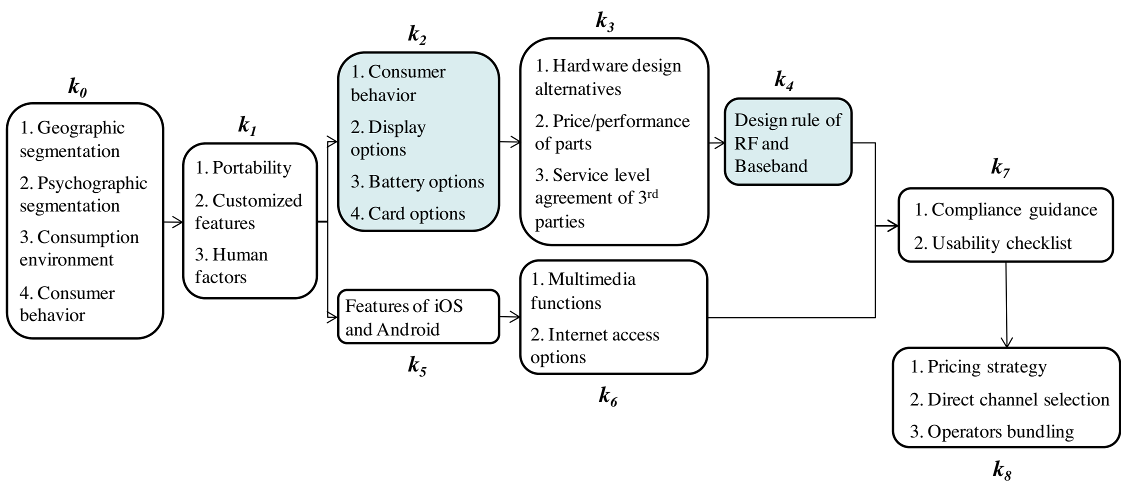

Fig. 9. Knowledge flow for the sourcing planners, where nodes k_2 and k_4 are essential knowledge nodes.

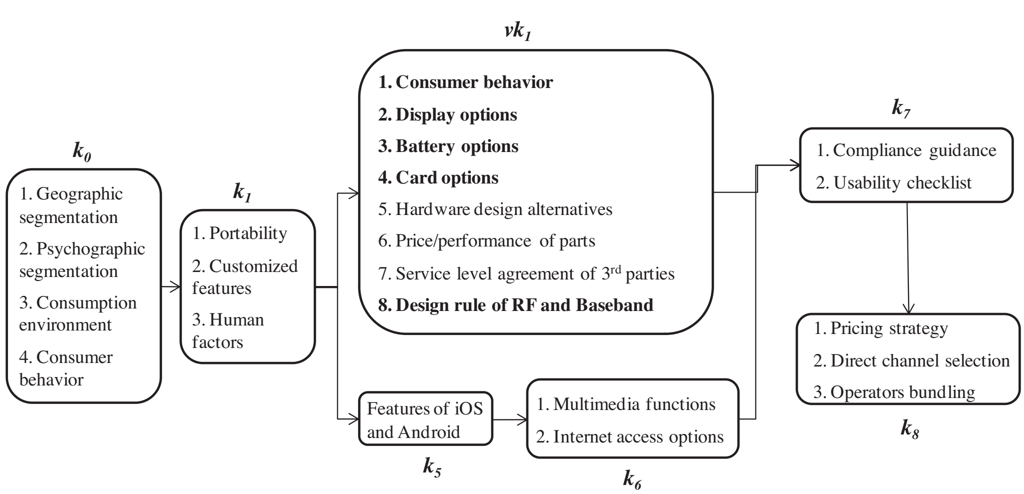

Fig. 10. A virtual knowledge node, vk_1, obtained after applying order-preserving approach.

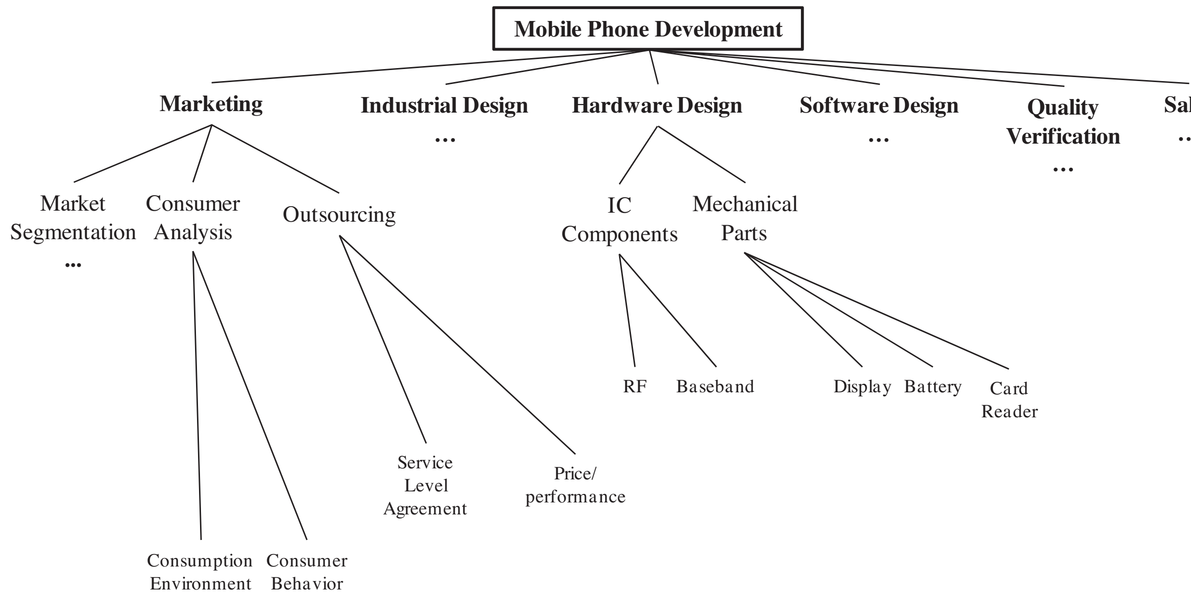

Fig. 11. A partial domain ontology of mobile phone development.

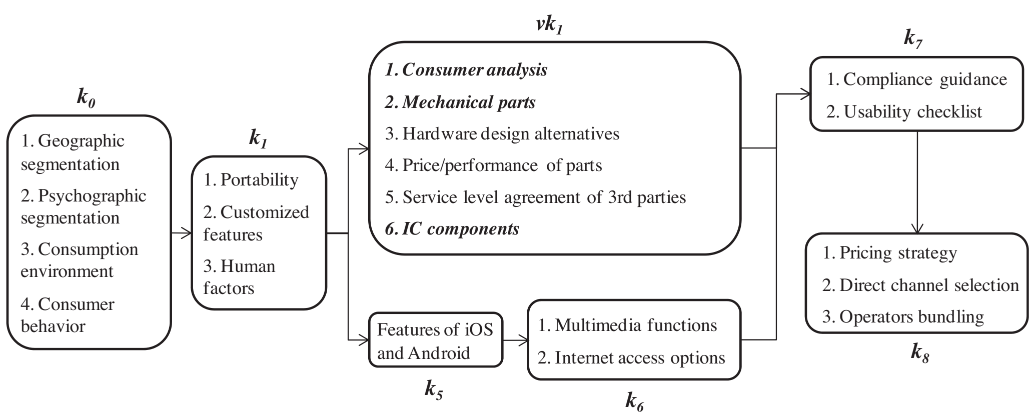

Fig. 12. Knowledge concepts of vk_1 after applying the minimum generalization policy.

bers with various knowledge-needs. Finally, every team member can obtain a proper knowledge-flow view to support his/her knowledge-needs in the collaborative knowledge support platform.

To test the practical implications of this study, a preliminary analysis was conducted. Several professionals were invited to examine the case and related concepts to investigate whether the theoretical model could benefit them. Overall, there was general agreement regarding the feasibility of the KFV model and its practical value. They thought that the KFV model could enhance typical knowledge flows to serve all team participants with their various knowledge-needs. For example, an interviewee mentioned that he would be able to communicate with a hardware designer more efficiently because the ontology of the theoretical model provided a common understanding of the general knowledge of hardware design. By referring to their different knowledge-flow views, both participants would be able to better understand their different knowledge-needs. Such understanding has the potential to improve the quality of communication and increase the efficiency of knowledge sharing.

The results of the preliminary analysis are summarized as follows: (1) the visualized knowledge flows and knowledge-flow views help team members to easily formulate their knowledge-needs and quickly obtain a consensus under a common domain ontology. Thus, the quality of their communication and decision making is improved; (2) knowledge-flow modelers can produce concealing criteria to protect confidential knowledge from unauthorized access and solve the information overload problem by abstracting detailed knowledge; and (3) in organizations, knowledge-flow views extend the efficiency of knowledge flows and improve the effectiveness of knowledge sharing and knowledge support.

5.1. Implications and discussion

This study contributes to knowledge management development, first by showing how a knowledge flow can address knowledge-needs. In the literature, models that formally illustrate both a knowledge flow and the corresponding knowledge-needs of workers are lacking. The proposed knowledge-flow model in Section 3 fills this gap by including three initiatives: (1) it adopts domain ontology to describe knowledge-needs by a composition of knowledge concepts; (2) it derives knowledge nodes from the activities in processes to visually display workers' knowledge-needs; and

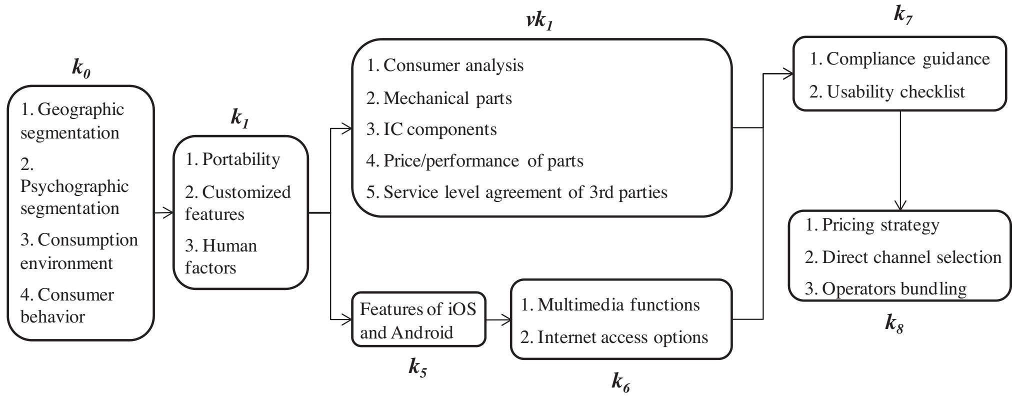

Fig. 13. The final knowledge concepts of vk_1 after removing implied knowledge concept.

(3) it defines flow dependencies as the sequence of an individual's or group members' knowledge-needs and/or the order of referencing codified documents. The knowledge-flow model helps researchers to obtain a clear view of knowledge-flow research.

Additionally, this study investigates the shortage of knowledge support in collaborative teams. The workers in a team usually have different knowledge-needs according to their task functions. For example, the mobile phone development process illustrated in Section 5 involves six task functions. The workers in these task functions need to access different knowledge concepts at different conceptual levels to conduct their work and communicate with each other. However, conventional knowledge-flow models do not provide the different views of a knowledge flow that are required to address individual needs. In Section 4 of this study, we propose a knowledge-flow view model to meet this and related challenges. According to the proposed model, knowledge-flow modelers select some knowledge nodes from a base knowledge flow to generate virtual knowledge nodes that conceal confidential or detailed information. Through an order-preserving approach and a knowledge concept generalization mechanism, a knowledge-flow view is generated. The proposed innovative model allows various knowledge-flow views to be generated that meet the individual knowledge-needs of different workers. These knowledge-flow views not only comply with organizational information security policy but also reflect the granularity of knowledge-needs. Thus, this study advances the conceptual applicability of knowledge flow research to cooperative knowledge support environments.

Practical implications can be derived from both the proposed knowledge-flow model and knowledge-flow view model. The knowledge-flow model can help organizations assess their current practices of knowledge sharing and knowledge reuse to gain insights into required knowledge concepts and build appropriate knowledge flows. The knowledge-flow view model can facilitate knowledge support in cooperative teams to improve team productivity and communication quality. Moreover, both models can be applied to any knowledge-based organization where business processes are conducted by cooperative teams in a dynamic working environment.

6. Conclusion and future work

In knowledge-intensive working environments, workers need task-relevant knowledge and documents to support task performance. To meet these requirements, many organizations have built knowledge support platforms that allow workers to preserve, share and reuse task-relevant knowledge. The value of knowledge support thus pertains to the importance of realizing knowledge management and promoting business intelligence in knowledge-based organizations.

Knowledge flow models have been proposed as an effective tool for building knowledge support platforms, and in recent years, a number of studies have focused on knowledge flow models and applications in business and scientific research contexts. Knowledge flows represent the flows of an individual's or group members' knowledge-needs and the referencing sequence of documents in conducting business operations and/or research activities. By depending on knowledge flows, organizations can facilitate their knowledge support mechanism by providing codified documents to workers that fulfill their knowledge-needs for individual tasks. However, since these task functions vary in a collaborative environment, different knowledge flow views are required. Conventional knowledge-flow models do not provide different views of a knowledge flow; this decreases the efficiency of knowledge sharing in organizations that depend on these models. To satisfy team workers with different knowledge-needs, this study proposes the knowledge-flow view model, which is capable of generating multiple knowledge-flow views.

The KFV model builds knowledge-flow views by abstracting knowledge nodes in a base knowledge flow to generate corresponding virtual knowledge nodes through an order-preserving approach and a knowledge concept generalization mechanism. The knowledge-flow views not only fulfill workers' different knowledge-needs but also facilitate knowledge support in teamwork. In summary, the KFV model advances the conceptual applicability of knowledge flow research to cooperative knowledge support environments and helps researchers to obtain a clear view of knowledge-flow research. It also improves the effectiveness of knowledge sharing and knowledge support in organizations.

6.1. Limitations and future work

One limitation of this work is the lack of a rigorous evaluation of the KFV model's practical benefits. Because this study constitutes fundamental knowledge flow research, it aimed to generate knowledge-flow views and extend knowledge flow research to cooperative teams by establishing a KFV model with novel methodology. The KFV model could be the core for building KFV systems based on the theoretical contributions achieved in this work. In the future, we will build a KFV system to realize the practical benefits of the KFV model and its related methodologies. An empirical study will also be conducted to quantify user satisfaction and business values by questionnaires or other measurement tools.

Another limitation of this study is that it does not consider how to integrate a workflow model. Organizations often adopt workflow models to manage the information in the business processes for task execution and team cooperation. In the future, we will propose to synthesize the information provided by workflow models to enhance the KFV model. This enhanced model will coordinate and formulate the interactions between work flows and knowledge-flow views to facilitate knowledge dissemination in knowledge-based organizations.

Furthermore, this work does not consider the diverse knowledge-needs of workers who play different roles and perform different operations in a team. Instead, this work focuses on defining a theoretical model of the knowledge-flow view and designing a kernel approach to derive knowledge-flow views from a base knowledge flow according to workers' concealing criteria. However, the concept of operations is essential in a real business environment. If a task involves teamwork, knowledge workers will play different roles when conducting different operations. Hence, they may require different conceptual levels of knowledge concepts in accordance with their roles and operations, which are denoted as workers' role-operation knowledge requirements. Extensive work is required to analyze and model the worker's role-operation knowledge requirements, and this is beyond the scope of the current work. Accordingly, our future work will develop a role-based KFV model from the perspectives of roles and operations.

Lastly, it is possible to obtain knowledge flows using other methods than the methods indicated in this paper. For example, knowledge flows might be derived from planning models generated by a knowledge-based planning methodology. In this way, the captured knowledge and derived constraints in the planning models would be the knowledge concepts of knowledge nodes. The dependency of planned tasks would establish the relationships between the knowledge nodes, forming a knowledge flow. Determining how to utilize such a knowledge-based planning methodology to generate knowledge flows systematically is another direction for future research.

Acknowledgements

This research was supported by the National Science Council of the Taiwan under grant NSC 99-2410-H-009-034-MY3.

References

[1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, M. Sintek, Context-aware, proactive delivery of task-specific information: the knowmore project, Information Systems Frontiers 2 (3) (2000) 253–276.
[2] A. Abecker, A. Bernardi, H. Maus, M. Sintek, C. Wenzel, Information supply for business processes: coupling workflow with document analysis and information retrieval, Knowledge-Based Systems 13 (5) (2000) 271–284.
[3] Y. Afacan, H. Demirkan, An ontology-based universal design knowledge support system, Knowledge-Based Systems 24 (4) (2011) 530–541.
[4] A. Anjewierden, R. de Hoog, R. Brussee, L. Efimova, "Detecting knowledge flows in weblogs", in Proceedings of the Thirteenth International Conference on Conceptual Structures, Atlanta (2005) 1–12.
[5] S. Biundo, R. Aylett, M. Beetz, D. Borrajo, A. Cesta, T. Grant, L. McCluskey, A. Milani, G. Verfaille, "Technological roadmap on AI planning and scheduling". PLANET, 2003.
[6] C. Chandra, A. Tumanyan, Organization and problem ontology for supply chain information support system, Data & Knowledge Engineering 61 (2) (2007) 263–280.
[7] I. Chebbi, S. Dustdar, S. Tata, The view-based approach to dynamic inter-organizational workflow cooperation, Data & Knowledge Engineering 56 (2) (2006) 139–173.
[8] H.H. Chen, C. Pang, Organizational forms for knowledge management in photovoltaic solar energy industry, Knowledge-Based Systems 23 (8) (2010) 924–933.
[9] S.Y. Choi, Y.S. Kang, H. Lee, The effects of socio-technical enablers on knowledge sharing: an exploratory examination, Journal of Information Science 34 (5) (2008) 742–754.
[10] H.K.H. Chow, K.L. Choy, W.B. Lee, A strategic knowledge-based planning system for freight forwarding industry, Expert Systems with Applications 33 (4) (2007) 936–954.
[11] P.F. Drucker, The coming of the new organization, Harvard Business Review 66 (1) (1988) 45–53.
[12] A.C. Edmondson, I.M. Nembhard, Product development and learning in project teams: the challenges are the benefits, Journal of Product Innovation Management 26 (2) (2009) 123–138.
[13] R. Eshuis, P. Grefen, Constructing customized process views, Data & Knowledge Engineering 64 (2) (2008) 419–438.
[14] T.R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (2) (1993) 199–220.
[15] M. Havey, Essential business process modeling, O'Reilly Media, Inc., 2005.
[16] J.J. Huang, The evolutionary perspective of knowledge creation - a mathematical representation, Knowledge-Based Systems 22 (6) (2009) 430–438.
[17] R.S. Huckman, B.R. Staats, D.M. Upton, Team familiarity, role experience, and performance. evidence from Indian software services, Management Science 55 (1) (2009) 85–100.
[18] Y.G. Ji, J.H. Park, C. Lee, M.H. Yun, A usability checklist for the usability evaluation of mobile phone user interface, International Journal of Human-Computer Interaction 20 (3) (2006) 207–231.
[19] J.-Y. Jung, K. Kim, D. Shin, J. Park, FlowWiki: A wiki based platform for ad hoc collaborative workflows, Knowledge-Based Systems 24 (1) (2011) 154–165.
[20] S. Kim, H. Hwang, E. Suh, A process-based approach to knowledge-flow analysis: a case study of a manufacturing firm, Knowledge and Process Management 10 (4) (2003) 260–276.
[21] J. Kingston, A. Macintosh, Knowledge management through multi-perspective modelling: representing and distributing organizational memory, Knowledge-Based Systems 13 (2–3) (2000) 121–131.
[22] C.-H. Lai, D.-R. Liu, Integrating knowledge flow mining and collaborative filtering to support document recommendation, Journal of Systems and Software 82 (12) (2009) 2023–2037.
[23] S.-T. Li, H.-C. Hsieh, Managing operation knowledge for the metal industry, Journal of Universal Computer Science 9 (6) (2003) 472–480.
[24] C. Lin, B. Tan, S. Chang, An exploratory model of knowledge flow barriers within healthcare organizations, Information & Management 45 (5) (2008) 331–339.
[25] H.-F. Lin, A stage model of knowledge management: an empirical investigation of process and effectiveness, Journal of Information Science 33 (6) (2007) 643–659.
[26] D.-R. Liu, C.-H. Lai, Mining group-based knowledge flows for sharing task knowledge, Decision Support Systems 50 (2) (2011) 370–386.
[27] D.-R. Liu, M. Shen, Workflow modeling for virtual processes: an order-preserving process-view approach, Information Systems 28 (6) (2003) 505–532.
[28] D.-R. Liu, I.-C. Wu, Collaborative relevance assessment for task-based knowledge support, Decision Support Systems 44 (2) (2008) 524–543.
[29] P. Liu, B. Raahemi, M. Benyoucef, Knowledge sharing in dynamic virtual enterprises: A socio-technological perspective, Knowledge-Based Systems 24 (3) (2011) 427–443.
[30] X. Luo, Q. Hu, W. Xu, Z. Yu, Discovery of textual knowledge flow based on the management of knowledge maps, Concurrency and Computation: Practice and Experience 20 (15) (2008) 1791–1806.
[31] C. Marinica, F. Guillet, Knowledge-based interactive postmining of association rules using ontologies, IEEE Transactions on Knowledge and Data Engineering 22 (6) (2010) 784–797.
[32] M.E. Nissen, An extended model of knowledge-flow dynamics, Communications of the Association for Information Systems 8 (2002) 251–266.
[33] V.T. Nunes, F.M. Santoro, M.R.S. Borges, A context-based model for knowledge management embodied in work processes, Information Sciences 179 (15) (2009) 2538–2554.
[34] A. Öhgren, K. Sandkuhl, "Towards a methodology for ontology development in small and medium-sized enterprises," in*IADIS Conference on Applied Computing*, Algarve, Portugal (2005).
[35] Á.E. Prieto, A. Lozano-Tello, Use of ontologies as representation support of workflows oriented to administrative management, Journal of Network and Systems Management 17 (3) (2009) 309–325.
[36] M.D. R-Moreno, D. Borrajo, A. Cesta, A. Oddi, Integrating planning and scheduling in workflow domains, Expert Systems with Applications 33 (2) (2007) 389–406.
[37] Y. Ren, K.M. Carley, L. Argote, The contingent effects of transactive memory: when is it more beneficial to know what others know?, Management Science 52 (5) (2006) 671–682
[38] A. Riege, Three-dozen knowledge-sharing barriers managers must consider, Journal of Knowledge Management 9 (3) (2005) 18–35.
[39] O.M. Rodríguez-Elias, A.I. Martínez-García, A. Vizcaíno, J. Favela, M. Piattini, A framework to analyze information systems as knowledge flow facilitators, Information and Software Technology 50 (6) (2008) 481–498.
[40] V. Sambamurthy, M. Subramani, Special issue on information technologies and knowledge management, MIS Quarterly 29 (1) (2005) 1–7.
[41] I. Savvas, N. Bassiliades, A process-oriented ontology-based knowledge management system for facilitating operational procedures in public administration, Expert Systems with Applications 36 (3) (2009) 4467–4478.
[42] K.A. Schulz, M.E. Orlowska, Facilitating cross-organisational workflows with a workflow view approach, Data & Knowledge Engineering 51 (1) (2004) 109–147.
[43] V. Sugumaran, V.C. Storey, Ontologies for conceptual modeling: their creation, use, and management, Data & Knowledge Engineering 42 (3) (2002) 251–271.

[44] Z. S. Syed, T. Finin, A. J., "Wikipedia as an ontology for describing documents", in Proceedings of the Second International Conference on Weblogs and Social Media, Seattle, Washington (2008).
[45] W.-T. Wang, S. Belardo, The role of knowledge management in achieving effective crisis management: a case study, Journal of Information Science 35 (6) (2009) 635–659.
[46] Y.-F. Wen, An effectiveness measurement model for knowledge management, Knowledge-Based Systems 22 (5) (2009) 363–367.
[47] S.-S. Weng, H.-L. Chang, Using ontology network analysis for research document recommendation, Expert Systems with Applications 34 (3) (2008) 1857–1869.
[48] M. Weske, Business process management: concepts, languages, architectures, Springer, 2007.
[49] I.-C. Wu, D.-R. Liu, P.-C. Chang, Learning dynamic information needs: a collaborative topic variation inspection approach, Journal of the American Society for Information Science and Technology 60 (12) (2009) 2430–2451.
[50] I.C. Wu, D.R. Liu, P.C. Chang, Toward incorporating a task-stage identification technique into the long-term document support process, Information Processing & Management 44 (5) (2008) 1649–1672.
[51] Y. Xu, A. Bernard, Quantifying the value of knowledge within the context of product development, Knowledge-Based Systems 24 (1) (2011) 166–175.
[52] C.C. Yang, J. Lin, C.-P. Wei, Retaining knowledge for document management: category-tree integration by exploiting category relationships and hierarchical structures, Journal of the American Society for Information Science and Technology 61 (7) (2010) 1313–1331.
[53] Z. Zhang, Z. Yang, Q. Liu, Modeling knowledge flow using Petri Net, in: IEEE International Symposium on Knowledge Acquisition and Modeling Workshop, China, 2008, pp. 142–146.
[54] W. Zhao, 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.
[55] H. Zhuge, Knowledge flow management for distributed team software development, Knowledge-Based Systems 15 (8) (2002) 465–471.
[56] H. Zhuge, A knowledge flow model for peer-to-peer team knowledge sharing and management, Expert Systems with Applications 23 (1) (2002) 23–30.
[57] H. Zhuge, Discovery of knowledge flow in science, Communications of the ACM 49 (5) (2006) 101–107.
[58] H. Zhuge, Knowledge flow network planning and simulation, Decision Support Systems 42 (2) (2006) 571–592.