Automating Knowledge Flows by Integrating Workflow and Knowledge Discovery Techniques

Surendra Sarnikar and J. Leon Zhao
Department of MIS, Eller College of Management
University of Arizona, Tucson, AZ 85721
{sarnikar, lzhao}@eller.arizona.edu

Abstract

Although it is apparent that knowledge flows should be automated to improve efficiency and effectiveness of knowledge sharing within an organization, there exist no frameworks or systems designed specifically for such a purpose. While there are a number of systems that can facilitate knowledge management, those systems do not support directly the modeling of knowledge flow and their systematic execution. In this paper, we investigate the research and technology gaps between the needs of organizational knowledge flow and the conventional workflow paradigm. We argue that conventional workflow cannot be used to automate knowledge flow directly because of a paradigm mismatch. Then, we propose a solution to this problem by outlining a system architecture that integrates knowledge discovery techniques and an intelligent workflow engine, leading to what we refer to as the knowledge workflow management system.

1 Introduction

Knowledge flow refers to the transfer of knowledge between knowledge workers across space, time and organizational boundaries and includes a sequence of tasks that need to be executed in order to enable this transfer of knowledge. Although it is apparent that knowledge flows should be automated to improve efficiency and effectiveness of knowledge sharing within an organization, there exist no frameworks or systems designed specifically for such a purpose.

Efficient knowledge flow is essential to the competitive survival of an organization (Gupta and Govindarajan, 2000). While a number of high-level workflow centric knowledge management frameworks (Stein and Zwass, 1995; Zhao, 1998), document recommender systems (Abecker et al., 2000; Kwan and Subramanian, 2003) and analysis methods (Nissen, 2002; Kim et al., 2003) have been proposed, they do not directly support the modeling of knowledge
workflows and their systematic execution. This technology gap is one of the major deterrents to knowledge sharing within an organization (Bruno, 2002). In order to automate knowledge flows efficiently, we introduce a new type of workflow called knowledge-workflow. A knowledge-workflow is a formal representation of a knowledge flow and will enable the systematic management of knowledge flows in an organization.

The rest of this paper is structured as follows. In Section 2, we present a scenario that illustrates a knowledge flow and the inefficiencies associated with current mechanisms for enabling knowledge flows. In Section 3, we present an overview of relevant work and identify the key technology gaps in this area. We present the knowledge workflow approach to automating knowledge flow in Section 4 and discuss the limitations of conventional workflow engines in executing knowledge workflows in Section 5. We present a formal framework for a knowledge workflow management system in Section 6 and conclude in Section 7.

2 An Example of Knowledge Workflow

Consider the knowledge flow scenario in which a consultant at a large knowledge-based firm needs to request some unique knowledge about a technology and its application. The consultant
can (1) search relevant knowledge repositories, (2) send request to a peer who is known to be able to answer the request, and (3) broadcast the request on a large mailing list. In order to satisfy the knowledge requirement, a consultant may need to execute varying combinations of all the above alternatives. A sample sequence of activities is given in Figure 1.

Further, consider the third sub-process which involves the use of a list server based mechanism for contacting experts. In this sub-process, the user sends the request to a list server which automatically forwards the request to all the experts subscribed to the list. The request may be labeled as relevant or non relevant to the expert depending on the match between the request and the expert’s area of interest. A graphical illustration of the list server based knowledge flow is given in Figure 2.

The manual execution of the knowledge workflow, which includes the use of a plain list server results in several inefficiencies and problems of information overload. For example, (1) the consultant may receive a large number of responses, several of which could be duplicate, (2) an expert may expend time on responding to a request that has already been satisfied, thus leading to wastage of resources, (3) the consultant may keep receiving responses even after the
knowledge requirement is satisfied (4) in a sequential flow, the consultant has to consecutively search through each repository when they can be simultaneously searched thus saving time, and (5) the experts may receive requests through the list server in which they are not interested, contributing to information overload.

Furthermore, on receiving responses from the experts the user may intend to continue the discussion with select users using an alternative tool such as a discussion board or a chat mechanism. In which case, the user has to manually transition to the next tool by creating appropriate instances and coordinating with select users to continue participation in the alternative tool. Currently there are no mechanisms to automate the tasks involved in the knowledge flows as described above. In the next section, we discuss the state of the art in knowledge flow modeling and support and identify research gaps that we intend to address in this paper.

3 Literature Review

Knowledge flow models have been presented from varying perspectives and at different conceptual levels in literature. Nissen (2002) proposes a dynamic model for classifying the knowledge flow patterns in an enterprise. The model is based on a characterization of the flow of work as horizontal processes and the flow of knowledge as complementary vertical processes. The vertical processes are cross-process activities that drive the flow of knowledge across space-time and organizational divisions in an enterprise. Ibrahim and Nissen (2003) apply the dynamic knowledge flow model to analyze complex knowledge flows in the construction industry during the feasibility-entitlement phase of a construction project. Zhuge (2002) proposes a knowledge flow model to enable knowledge sharing between peers in a team environment. The model involves a two-dimensional knowledge field where knowledge is represented along the
dimensions of knowledge type and knowledge level. The model is based on the assumptions that the knowledge requirements of similar tasks are of the same type and the knowledge requirements of peers are at the same knowledge level.

A methodology for document flow coordination is presented in Zhao (2002). The approach involves the creation of an Organizational Knowledge Network, which consists of a work-nets and awareness nets. An analytical method called the Knowledge Association Algebra is also presented for maintaining and initializing the Organizational Knowledge Network. Kim et al., (2003) propose a process-based framework for analyzing knowledge flow. They identify different knowledge flow patterns and present a technique to document knowledge at the conceptual, logical and physical levels. They propose a six-step approach to knowledge flow analysis that includes defining an ontology, process analysis, knowledge extraction, knowledge flow analysis, knowledge specification, and knowledge validation.

In addition to knowledge flow modeling and analysis, a related category of literature includes workflow systems that are extended to model a wider range of processes. Various approaches have been explored in imparting flexibility to workflow systems. A related concept is of ad-hoc workflows (Voorhoeve and van der Aalst, 1997). An ad-hoc workflow enables end users to modify processes during execution. Ad-hoc workflows are based on process templates that form a part of a hierarchy. While higher-level templates are relatively inflexible, the lower level templates can be modified based on the requirements of each case.

Some of the conceptual foundations of knowledge flow automation can be traced to high-level frameworks for workflow enabled knowledge management systems. Stein and Zwass (1995) propose a high level framework for a new type of information system called the organizational memory information system (OMIS), which is aimed towards acquisition,
retention, search and retrieval of knowledge. Workflow systems are well suited for actualizing organizational memory (Zhao, 1998) and can serve as a conduit for knowledge distribution and management (Zhao et al., 2000). The frameworks proposed by Stein and Zwass (1995) and Zhao (1998) describe the requirements and architecture of IT enabled knowledge flow systems at a high level of abstraction and help guide future research in this area. However, they do not contain sufficient detail to guide the actual implementation of an enterprise-wide knowledge flow infrastructure.

Although high-level workflow centric knowledge management frameworks (Stein and Zwass, 1995; Zhao, 1998), document recommender systems (Abecker et al., 2000; Kwan and Subramanian, 2003) and analysis methods (Nissen, 2002; Kim et al., 2003) have been proposed, they do not support directly the modeling of knowledge flows and their systematic execution. This technology gap is one of the major deterrents to knowledge sharing within an organization (Bruno, 2002).

The automatic execution of some of the tasks in knowledge flows can be supported by individual systems such as expert locator system (McDonald and Ackerman, 1998) summarization systems (Moens et al., 2005) and filtering systems. However, each of these systems has to be individually invoked by the user to accomplish the knowledge goal. Current literature is lacking in a framework for automatically integrating and executing such tasks for achieving the user’s knowledge goal and improving the efficiency of knowledge flows.

A knowledge flow infrastructure is especially important to knowledge intensive organizations such as consulting firms. Some consulting firms have developed customized applications to enable the management of organizational knowledge. Typical examples of such systems include the Knowledge Xchange at Anderson Consulting, and the Knowledge On-Line
system at Booz Hamilton. These systems provide a suite of applications such as expert location, knowledge repositories, discussion boards and templates for codifying knowledge (Garcia, 1997; Tristram, 1998).

Although such customized applications can automate some knowledge flows, they are specific to an organization and its business processes. Developing an implementation that is flexible and can support a wide variety of knowledge flows requires the development of formal methodology that can help model the knowledge flows, and the extension of current workflow technologies to handle the knowledge flows that are characterized by dynamically changing models. The framework and formalisms we propose in this paper lays the foundation for such a methodology.

4 The Automation of Knowledge Workflow

In order to alleviate the above problems, improve knowledge flow efficiency and better manage the knowledge flows across an organization, we propose the automation of knowledge flows using a knowledge workflow. The automation of knowledge flow requires the integration of information retrieval mechanisms with workflow systems. For example, consider the list server based knowledge flow when enhanced with intelligent retrieval based services as shown in Figure 3.

In the plain list server based knowledge flow, several experts (experts C and D in Figure 2) that are contacted turn out to be uninterested in the message. The user is also overwhelmed with multiple, possibly duplicate, responses. In the intelligent services enhanced knowledge flow, the use of a filtering service (Sarnikar et al., 2004) denoted by “F” prevents the distribution of the message to non relevant experts. The use of summarization and aggregation services (Moens et al., 2005), denoted by “S” and “A”, prevents overloading the user.
Figure 3. A list server based knowledge flow enhanced with intelligent services

While information retrieval mechanisms provide discovery and matching services, workflow systems coordinate the invocation of the appropriate intelligent service and automate the routing and delivery of messages and documents. Specifically, given a set of user specified constraints, a workflow can be designed that can automatically invoke intelligent services, contact experts, retrieve document and present the results to the user.

For example, consider the knowledge request specified by a consultant in a large consulting firm as given in Figure 4. The consultant wishes to acquire information on healthcare data privacy compliance measures in the European market. Additionally, the user has also specified the sources of knowledge to search and the sequence in which to search the sources. The user also prefers to receive the knowledge in real-time as opposed to a batch process and requires the process to end on satisfaction of the knowledge flow.

Request: Information on healthcare data privacy compliance measures for Europe
Sources: 1. Internal Knowledgebase; 2. Domain Experts
Search: Parallel
Response: Continuous
Time limit: None
Close on satisfaction: Yes

Figure 4. User specified constraints for a knowledge workflow
We use the BPMN notation (OMG, 2006) to describe the knowledge workflow and introduce a new symbol to denote uncertain nodes in the process model called a model update point. The model update point in a workflow indicates that the node where a model needs to be updated based on either user input or the data generated during execution of a knowledge workflow. A summary of the notation used is given in Figure 5. The knowledge workflow corresponding to the above request is shown in Figure 6. The workflow is initiated by the consultant to satisfy a one time knowledge requirement specified by a request. In this workflow, the user receives messages in real time and continuously. The knowledge workflow is designed to terminate when the user receives the first relevant document. Note that the knowledge workflow consists of four separate workflow sub-models that collaborate together to accomplish the workflow. We refer to these sub-models as knowledge workflow patterns\(^1\), which can be assembled at the runtime to support knowledge workflows (Sarnikar and Zhao 2006).

\[\text{Start} \quad \text{Activity} \quad \text{End} \quad \text{Sub-process} \quad \text{OR-Split and Merge} \quad \text{Model update point} \quad \text{AND-Split and Join} \quad \text{Control Flow} \quad \text{Event-based OR split} \quad \text{Message flow}\]

\[\text{Figure 5. BPMN and other notation}\]

\(^1\) Note that knowledge workflow patterns are different from workflow patterns (Aalst et al. 2003). The latter are elementary syntactic patterns found in workflow models, but knowledge workflow patterns have semantic meaning.
The knowledge workflow is initiated when the consultant sends a query document to the knowledge workflow management system (KWMS). The system invokes the intelligent knowledge recommendation system (IR System) to identify relevant experts and repositories and routes the query to appropriate experts and repositories. On receipt of a response, the system routes the response to the initiator of the knowledge workflow. The initiator reviews the responses as they are received and closes the knowledge flow as soon as the knowledge requirement is satisfied.

Figure 6. A workflow for executing one-time knowledge flow

The process components displayed in the expert’s pool in Figure 6 present the process from the domain expert’s perspective. The domain expert’s process component begins when his or her email or knowledge management system receives the knowledge request. If the knowledge
workflow is closed before the expert opens the request, the request is deleted and the flow is
terminated, otherwise the expert can open the request and respond to the request. The pool
labeled knowledge workflow system illustrates the process components executed by the KWMS.
It involves the coordination of activities of the requester, the experts, and systems such as the
expert locator and information retrieval system.

The automation of the knowledge flows using the knowledge workflow illustrated above has
the following advantages. (1) It saves time by simultaneously initiating multiple resources to
respond to the knowledge query. (2) It prevents overloading the user with responses even after
the satisfaction of the request by terminating the knowledge flow. (3) The termination of the
knowledge flow also prevents the domain experts from responding to requests that have already
been satisfied. (4) The integration with information retrieval and filtering services via an
intelligent recommendation system (IR System) alleviates the information overload problem by
preventing the routing of request to experts that are not relevant to the given query, and by
filtering out duplicate responses from experts and repositories. (5) It enables knowledge
codification and reuse by automatically saving the query response pairs to a knowledgebase.

The knowledge workflow needs to be created step by step during the runtime as follows.
First, the user makes a request for knowledge according the specification in Figure 4. An
appropriate knowledge workflow model such as the one found in the first box, i.e., a pool in
BPMN notation, in Figure 6. This model needs to be started. Then, the second sub-model will
be triggered. At sometime later, the third sub-model will be triggered based on the result from
the second sub-model. Similarly, the fourth sub-model will be triggered as well. Two important
concepts need to be explained here. First, these workflow patterns are difficult to assemble a
priori since many different combinations of patterns exist. Second, the parameters of the patterns
cannot be determined at the start of the runtime since they need to be determined via knowledge
discovery based on knowledge relevancy. These two unique features of knowledge workflow
management make it difficult to automate within the confinement of a conventional workflow
management system.

In the example above, notice that the sub-workflow corresponding to the knowledge workflow
management system also includes a monitoring sub-process following which the knowledge
workflow is either closed or extended with an updated model. A corresponding update model activity requiring user input is included in the sub-workflow depicted in the User pool. The update model task is an integral part of knowledge workflows and is triggered towards the end of a sub-workflow from either a user or system perspective, or whenever an exception is generated following a monitoring sub-process.

In total, the above process has two points of uncertainty which require updates to the model during execution. The first point of uncertainty is denoted by the model update point symbol in the knowledge workflow system pool. This uncertainty is due to the uncertain number of AND splits when sending requests to the experts and repositories. The AND split in this case can be substantiated based on the expert locator output that is generated during workflow execution. Furthermore, the monitor sub-process may loop back to the model update point as the model may need to be update during execution due changes in the availability states of the experts.

The second model update point occurs in the user’s pool following the decision node. On reviewing the responses, the user may be satisfied with the response or may choose between reviewing additional responses and further discussing the responses with the experts. Since the mechanism of discussion and the number of participants is not uncertain, this sub-process needs
to be determined at run-time and is represented by the model update point. Following the sub-
process the user may be satisfied with the response or may reformulate the request.

Considering the uncertain nature of knowledge flows, the capability to update models during
the execution of knowledge workflows is a key requirement for the automation of knowledge
flows. In the above illustrated knowledge flow, upon receipt of a few responses a user could
choose among several different courses of action including modifying the knowledge request,
resending it to a specific subset of experts or initiating a discussion with respondents using a
particular discussion tool such as an electronic meeting system. Each of the above courses of
action would require the extension of the current workflow model with a new sequence of tasks.
Given the large number of possibilities for model extension, it is not feasible to develop a
comprehensive process model for knowledge flows. Hence, a dynamic model update mechanism
is a key requirement for knowledge workflows.

In the next section, we describe a conventional workflow management system and describe
its limitations in executing knowledge workflows. In Section 6, we describe a formal framework
for knowledge workflow management and describe its key features which enable the execution
of knowledge workflows, including the dynamic updating of models.

5 Limitations of the Conventional Workflow Paradigm

While knowledge workflows are similar to structured business processes in some aspects,
key differences prevent the modeling and execution of knowledge workflows using the existing
workflow management systems. First, in a typical business process such as order processing, the
control flow and data flow are predetermined while knowledge flow is of ad hoc nature and
evolves based on user requirements and system constraints that are difficult to define a priori.
Second, with typical workflow management systems, activities in a business process are assigned
to roles, which are then resolved at run time. However, in knowledge flows, the role-based workflow paradigm breaks down since the routing of activities and flow of documents in knowledge flow are based on retrieval-based matching criteria as opposed to role resolution.

Figure 7. The conventional workflow paradigm (Wfmc, 1999)

The conventional workflow paradigm, as illustrated in Figure 7, requires that the workflow model be “execution ready” before it is deployed into the workflow engine. The prevalent workflow management systems do not allow the change of workflow models once its execution starts.

6 A Formal Framework for Knowledge Workflow Management

In order to execute the knowledge workflows, we propose a component-based architecture for knowledge workflow management. The knowledge workflow management system can interact (1) with users and groups within a company and consist of four major system components: (2) knowledge workflow modeler, (3) intelligent workflow engine, (4) intelligent expertise locator and (4) intelligent document recommender.
The information retrieval engine is used to execute functions such as document recommendation, aggregation, filtering and other retrieval related functions. The intelligent expertise locator is used for a function that is analogous to the role-resolution function in traditional workflow systems. The former uses multiple sources of information including organizational hierarchies, user interest profiles and social network analysis to identify relevant experts. The knowledge workflow modeler assembles knowledge workflow patterns to develop a knowledge workflow for satisfying the given requirement. The knowledge workflow is executed using a state-machine based workflow engine (Apache Software Foundation, 2006).

We use an intelligent workflow engine as it is better suited for executing workflows where the control flow and sequence of activities cannot be determined at design time, and is based on the outcome of intermediate events and input from user. In the proposed architecture, the knowledge workflow management system relies on two different perspectives of a knowledge flows, a control flow perspective and a state machine perspective. We specify formally the
various system components (See Figure 8) and related concepts. These specifications will be used as the basis for various types of system analysis.

**Definition 1 - Knowledge Workflow Pattern.** We define a knowledge workflow pattern as a 5-tuple \( P = \langle A, C, R, D, AS \rangle \), where

- \( A \) is a set of activities, each of which is assigned to a specific resource
- \( C \) is a set of conditions
- \( R \) is a set of resources and includes users, groups and machines
- \( An \) is a set of assignments, \( An \subseteq A \times R \) where each activity is assigned to a predetermined resource.
- \( D \) is a set of data items associated with the workflow
- \( AS \) is an ordering of activities where \( AS \subseteq A \times A \times C \)

**Definition 2 - Knowledge Workflow Model.** A knowledge workflow model is defined as follows \( M = \langle P, An \rangle \), where

- \( P \) is a knowledge workflow pattern, and
- \( An \) is a set of assignments, \( An \subseteq A \times R \) where each activity is assigned to a predetermined resource.

**Definition 3 - State Machine Workflow.** A workflow state machine is defined as \( W = \langle M, S, E, T \rangle \), where

- \( M \) is a knowledge workflow model
- \( S \) is a set of allowable states for the knowledge workflow
- \( E \) is a set of events that trigger a change in the state of the knowledge workflow
- \( T \) is a set of transitions where \( T \subseteq E \times S \)
The state chart depicting the various states of a knowledge workflow model is given in Figure 9. The states of the process instance changes based on the events generated on execution of the sub-workflows that compose the knowledge workflow.

**Definition 4 - Intelligent Expertise Locator.** The intelligent resource locator is used to identify experts relevant to the given query. Specifically, the Intelligent Resource Locator provides two basic functions, function $f$ to identify relevant experts and function $g$ to identify relevant knowledge-bases. They are defined as follows.

\[
R_H = f(q, R_H)
\]

\[
R_M = g(q, R_M)
\]

![Figure 9. Process instance states](image)

**Definition 5 - Knowledge Workflow Instance.** A knowledge workflow instance is given by

\[ I = <W, s, i> \], where

- $W$ is a knowledge workflow model
- $s$ is the state of the knowledge workflow and
• *i* is an instantiation of the data items in the workflow model.

The various states of a process instance are depicted in Figure 9. We extend the state-machine representation of a process instance of a conventional workflow system to include 4 additional states, *update trigger, model update, verification* and *model restarting* to account for the dynamic model update features of the knowledge workflow management system.

![Figure 10. A sequence diagram showing interaction between various components](image)

**Definition 6 - Knowledge Workflow Management System.** The knowledge workflow management system integrates all the above functions to provide a mechanism for automating knowledge distribution. The activities of the knowledge workflow management system can be summarized by the following algorithm.

Step 1. Receive query and parameters describing knowledge requirements

Step 2. Using the knowledge workflow modeler, identify suitable patterns

Step 3. Initialize pattern with users, using the intelligent resource locator

Step 4. Submit the knowledge workflows to the state machine workflow engine for execution.
Step 5. If the knowledge request is active on completion of the workflows, generate new knowledge workflows by repeating steps 1 through 4 above.

A sequence diagram illustrating the interaction between the components of the knowledge workflow management system is given in Figure 10.

7 Conclusions

In this paper, we argued that the conventional workflow paradigm is insufficient for supporting knowledge workflows directly due to a mismatch between the dynamic and in deterministic nature of knowledge workflow and the fixed model paradigm of conventional workflow engine. Further, we proposed an architecture for knowledge workflow management systems that consists of an intelligent workflow engine, a knowledge workflow modeler, an intelligent expertise locator, and an intelligent document recommender.

We also presented a set of mathematical representations of these components, which are used to describe their functionality. These mathematical representations help illuminate the theoretical underpinning of the system functions as shown also in a sequence diagram and allow further analysis in our future research as exemplified in the following topics:

- Verification of the completeness and correctness of the state-machine workflow representations that are generated from a knowledge workflow model
- Investigation of conflicts, exceptions, and other issues related to the dynamic extension of knowledge workflow models
- Identification of knowledge workflow patterns including basic patterns and their assembly towards more complex knowledge workflows.
- Prevention of information overload by controlling the intensity of knowledge flows at the user and system levels.
References


