

# 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 with an organization, there exists no frameworks or systems designed specifically for such a purpose. While there are a number of systems that facilitate knowledge management, those systems do not support directly the modeling of knowledge workflows and their systematic execution. In this paper, we investigate the research and technology gaps between the needs of organizational knowledge flows and the conventional workflow paradigm. We argue that the conventional workflow cannot be used to automate knowledge flows 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

Efficient knowledge flow is essential to the competitive survival of an organization (Gupta and Govindarajan, 2000). In order to improve the efficiency of knowledge flows, the knowledge flow activities can be modeled in terms of a workflow model which we refer to as knowledge workflows, and subsequently automated using a process automation system (Sarnikar and Zhao, 2006).

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 support directly 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 this paper, we argue that current workflow engines are not capable of handling knowledge workflows due to their instance-based and dynamic nature, and to address this technology gap, we propose a new architecture for a knowledge workflow management system. We provide a mathematical representation of various components of the knowledge workflow management system and demonstrate a state-machine based mechanism to enable the execution of knowledge workflows.

The rest of this paper is structured as follows. In Section 2, we present a knowledge workflow scenario and illustrate the deficiencies of conventional workflow systems in Section 3. In Section 4 we propose a knowledge workflow management system that extends the conventional workflow paradigm for executing knowledge workflows. We identify research issues and conclude in Section 5.

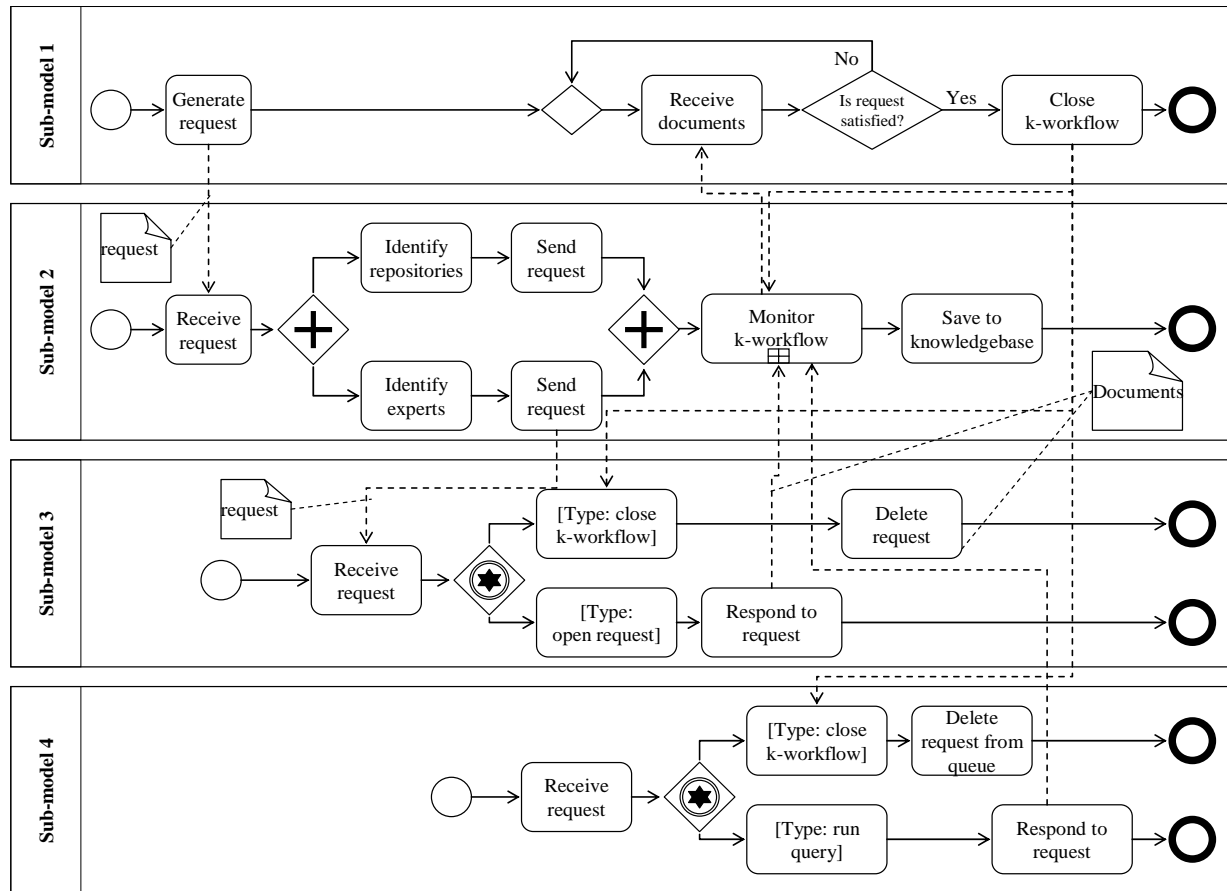
## 2. Example Knowledge Workflow Scenario

We present a scenario where a user requests knowledge in terms of documents from a repository and experts in an organization. The knowledge flow request is given in Figure 1.

<p><b>Request:</b> Information on healthcare data privacy compliance measures for German markets  <b>Sources:</b> 1. Internal Knowledgebase; 2. Domain Experts  <b>Search:</b> Parallel  <b>Response:</b> Continuous  <b>Time limit:</b> none  <b>Close on satisfaction:</b> yes</p>
--

**Figure 1. A Knowledge Workflow Requirements Specification**

A knowledge workflow satisfying the above specification is given in Figure 2, which is based on the BPMN notation (OMG, 2006). 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<sup>1</sup>, which can be assembled at the runtime to support knowledge workflows (Sarnikar and Zhao 2006).



**Figure 2. An Example Knowledge Workflow**

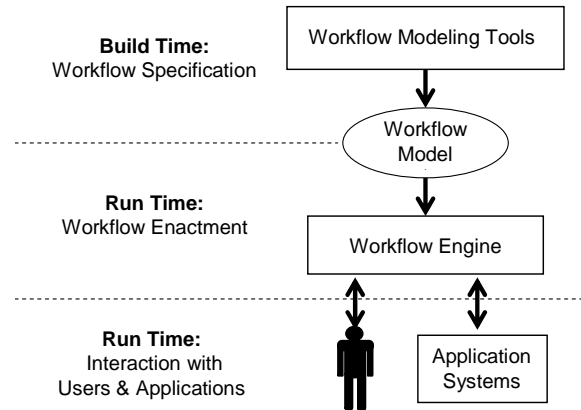
<sup>1</sup> Note that knowledge workflow patterns are different from workflow patterns (Aalst et al. 2003). The latter are elementary syntactic patterns found in workflow models, but knowledge workflow patterns have semantic meaning.

The knowledge workflow 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 1. An appropriate knowledge workflow model such as the one found in the first box, i.e., a pool in BPMN notation, in Figure 2. 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. As we explain in a later section of the paper, these two unique features of knowledge workflow management make it difficult to automate within the confinement of a conventional workflow management system.

### 3. A Formal Model of Conventional Workflow Paradigm

In order to understand the conventional workflow paradigm, we develop a state machine representation of workflow engine, which is the core of a typical workflow management system.



**Figure 3. The conventional workflow paradigm (WfMC 1999)**

As illustrated in Figure 3, the conventional workflow paradigm 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. We can refer to this paradigm as the “fixed model paradigm” and represent it mathematically by means of a conventional workflow state machine.

**Definition 1 - Conventional Workflow State Machine.**

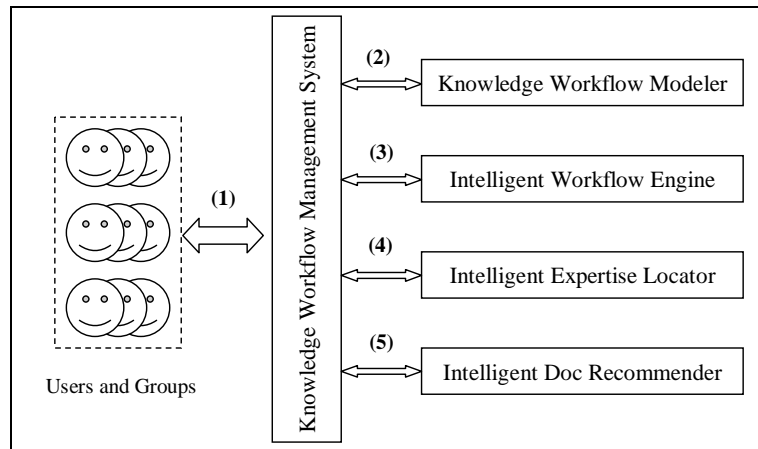
$W_C = \langle M, S, T \rangle$ , where

- $M$  is a workflow model
- $S$  is a set of allowable states for the workflow
- $T$  is a set of transitions where  $T \subseteq E \times S$

The conventional state machine  $W_C$  starts with a workflow model, instantiates it with all necessary parameters, makes transitions through the allowable workflow states, and eventually reaches an end state of the workflow. It is clear that the conventional workflow engine would not be able to handle knowledge workflows as illustrated with the knowledge workflow scenario

given in the previous section. We will show how the conventional workflow state machine can be extended in order to handle knowledge workflows.

#### 4. A Formal Framework for Knowledge Workflow Management



**Figure 4. Architecture for Knowledge Workflow Management System**

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 state machine based workflow executes an instance level model that is provided by the knowledge workflow management system.

We specify formally the various system components (See Figure 4) and related concepts. These specifications will be used as the basis for various types of system analysis.

**Definition 2 - 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 3 - Knowledge Workflow Model.** A knowledge workflow model is defined as follows

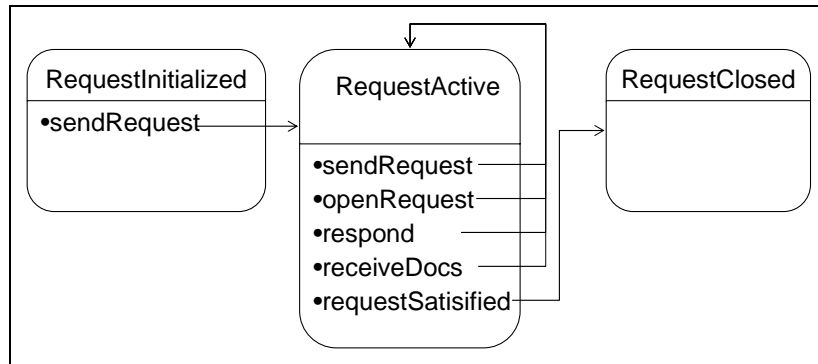
$M = \langle P, An \rangle$ , where

- $P$  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 4 - 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  $T \subseteq E \times S$



**Figure 5. A State Chart for the Sample Knowledge Workflow**

The state chart corresponding to the sub workflows described earlier is given in Figure 5. On execution of all the workflows, if the knowledge request remains unsatisfied, i.e. the query does not reach and end state of query closed, the workflow engine consults the knowledge workflow management system for a model update, where the user supplies either a new query or new constraints, based on which a new knowledge workflow model is generated for execution.

**Definition 5 - 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.

$$R_H^i = f(q, R_H)$$

$$R_M' = g(q, R_M)$$

**Definition 6 - Knowledge Workflow Instance.** A knowledge workflow instance is given by

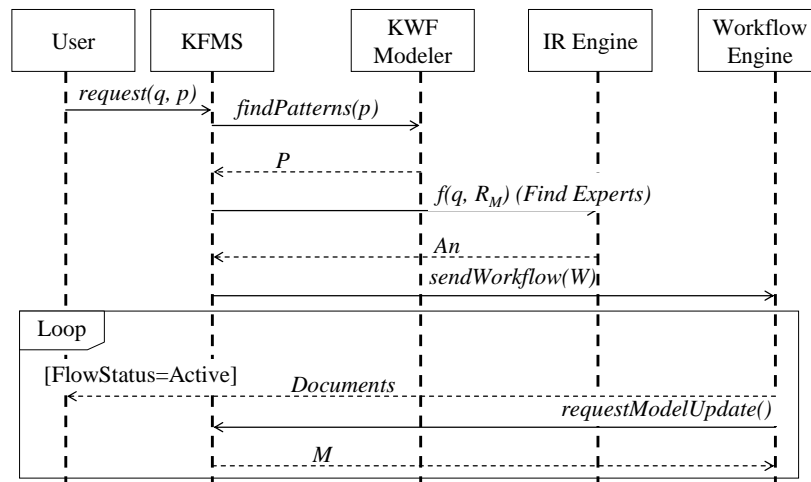
$I = \langle W, s, i \rangle$ , where

1.  $W$  is a knowledge workflow model
2.  $s$  is the state of the knowledge workflow and
3.  $i$  is an instantiation of the data items in the workflow model.

**Definition 7 - 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 6.



**Figure 6. A sequence diagram showing interaction between various components**

## 5. 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
- Investigation of access control issues to ensure security of the organizational knowledge, and
- Prevention of information overload by controlling the intensity of knowledge flows at the user and system levels.

## References

1. Aalst, van der, W.M.P. and Hofstede, (2003) "Workflow Patterns", *Distributed and Parallel Databases*, 14(1), 5-51.
2. Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn O. and Sintek, M. (2000), "Context-aware, proactive delivery of task-specific information: the KnowMore Project", *ISF*, 2, 3/4, 253–276.
3. Apache Software Foundation, "Apache Lenya 1.2 Documentation, The Workflow State Machine", July 31, 2006, <[http://lenya.apache.org/docs/1\\_2\\_x/components/workflow/state-machine.html](http://lenya.apache.org/docs/1_2_x/components/workflow/state-machine.html)>
4. Bruno, J (2002) "Facilitating Knowledge Flow through the Enterprise", *International Journal of Intelligent Systems in Accounting, Finance & Management*, 11, pp 1-8.
5. Gupta, A. and Govindarajan, V (2000) "Knowledge Flows within Multinational Corporations", *Strategic Management Journal*, 21 pp 473-796.
6. Kwan, M. and Balasubramanian, P. (2003) "KnowledgeScope: Managing Knowledge in Context", *Decision Support Systems* 35(4), 467-486.
7. OMG, "Business Process Modeling Notation Specification 1.0", Object Management Group, Needham, MA
8. Sarnikar, S. and Zhao, J.L (2006) "A Pattern-based Approach to Automating e-Business Knowledge Flows", *Proceedings of the Fifth Workshop on E-business (WeB 2006)*, Milwaukee, WI, December 9, 2006.
9. WfMC "Workflow Management Coalition Terminology & Glossary, WfMC-TC-1011, Issue 3.0," Workflow Management Coalition, Winchester, Hampshire, UK, p. 65.
10. Zhao, J. L. (1998) Knowledge Management and Organizational Learning in Workflow Systems, *Proceedings of the AIS Americas Conference on Information Systems*, Baltimore, Maryland.