Supporting Workflow Management Systems with Traceability

Peng Xu and Balasubramaniam Ramesh
Department of Computer Information Systems
Georgia State University
{pxu|bramesh}@gsu.edu

Abstract

Workflow Management Systems need to support organizations adapt their workflows to meet changing environments and business processes. We propose a traceability based knowledge management approach to support such a capability in WFMS. We present a framework for representing traceability knowledge to capture the context in which workflows are specified and evolved. We discuss the functionalities of a knowledge management system that can support dynamic reconfiguration of workflows in the context of changing business processes as well as for maintaining their integrity.

1 Introduction

Today’s enterprises must react quickly to changes, rapidly develop new services and products, and at the same time improve productivity and quality and reduce cost. In this process, Business Process Re-engineering (BPR) efforts can help businesses re-design their structures and processes. Among the variety of ways Information technology can be used to support this process, workflow management systems (WFMS) play a central role. Often organizations need to implement large and heterogeneous distributed systems where a set of interrelated tasks can be executed in efficient way. Workflow automation to coordinate activities throughout the enterprise is increasingly recognized as an important approach to support these requirements [1]. The WFMS can also play a vital role in managing knowledge of the enterprise, such as managing business rules [2]. As summarized in [3], the key benefits of WFMS are:

- Improved efficiency - automation of many business processes results in the elimination of many unnecessary steps
- Better process control - improved management of business processes achieved through standardizing working methods and the availability of audit trails
- Improved customer service – consistency in the processes leads to greater predictability in levels of response to customers
- Flexibility – software control over processes enables their re-design in line with changing business needs
- Business process improvement - focus on business processes leads to their streamlining and simplification

Proof of the ever-increasing interest in workflow management is the large number of commercial products that have appeared in the last few years including: Action Workflow System, of Action Technologies; IBM’s Flow Mark; Visual WorkFlow from FileNet; OPEN/workflow, a WANG’s product [4]. Workflow technology has become mainstream application-development tools and application-integration middleware that support the management of business process realization in a variety of application areas. Enterprise Resource Planning Systems (ERP) may be thought of as focusing on business applications of workflow management system [5]. ERP systems may be seen as “a set of more or less fine-grained business functions and a set of predefined workflows that realize complex business operations”. When the system is customized for a specific setting, workflows can be modified and new ones can be constructed [6].

In the context of BPR (business process re-engineering), workflow management provides a means for enacting reengineered processes and for gathering information about the actual performance of these processes.

As the consequence of rapidly changing environment, businesses need to reengineer business models and processes frequently. Those organizations in
the modern business world, which refuse to change, are headed toward rapid obsolescence because they cannot compete. These business realities need to be supported by a WFMS by providing the ability to rapidly and dynamically change and enact workflows.

The basic premise of our work is that WFMS can benefit from the ability to manage knowledge about how workflow components relate to business process as well as with each other so that the evolution of workflows when the business processes that they are intended to support evolve. Specifically, we advocate the use of traceability, the ability to follow the life of an artifact, as the primary facility to manage the knowledge about the evolution of workflows. In this paper, we present a framework for representing traceability knowledge and discuss tools to support the capture and use of this knowledge to support critical activities in WFMS. In the section 2, features of workflow management systems are introduced. We further discuss the role of knowledge management in supporting workflow management. Section 3 outlines the role of traceability in supporting complex organizational processes. A case is discussed in section 4. A traceability framework to support workflow design and maintenance is proposed in section 5. Section 6 discusses the features of a knowledge management system to illustrate its functionalities. This is followed by discussion of related work and future work.

2 Workflow Management Systems (WFMS)

Workflow management is defined as the management of processes through the execution of software whose order of execution is controlled by a computerized representation of the process. The primary reason for the popularity of workflow technology is its support for the management trends including reinvention and revitalizing corporations through rightsizing and business-process reengineering. Current implementations of workflow systems such as Enterprise Resource Planning (ERP) systems automate core corporate activities and let companies share common data and practices across the enterprises. Workflow systems are designed to assist groups of people in carrying out work procedures, and contain organizational knowledge of where work flows. Workflow systems are defined as “systems that help organizations to specify, execute, monitor, and coordinate the flow of work items in a distributed environment” [7]. A WFMS provides the software tools to define, manage and execute workflows. WFMS have two main functions: a build time function and a run time function. Build time functions help administer workflow process and run time interactions with workflow users and software applications [8].

Most current workflow management systems are static systems, which means to reconfigure WFMSs, systems have to be stopped and re-instantiated. Given the dynamic structure of today’s organizations, it is unlikely that business processes are modeled once and executed without any change. More flexibility is desired in WFMSs. The adaptive or dynamic workflow systems are regarded as efficient solutions to the problem [9], [10]. Dynamic Workflow Management Systems (DWFMS) allow users to change the data and the structure of the workflow on the fly. System evolution is unavoidable because business processes evolve continuously caused by internal organizational reforms and external environment changes. The static WFMS cannot be efficient in such environments. There is an increasing demand for dynamic adaptive workflow management systems, which can deal with dynamic changing.

Management of the dynamics of workflow configurations is required in a variety of application domains. Consider a manufacturing workflow system that can process orders, schedule processes, control material orders, calculate bills of material, and track orders and products. Any change in manufacturing process necessitated by changes in environmental conditions need to be reflected in the systems. For example, facing changing demands of the market, the manufacturer may decide to outsource the production of some parts rather than making them in-house. The WFMS needs to support such dynamically changing workflows to accommodate the changing of the business processes.

3 Traceability

Traceability refers to the ability to describe and follow the life of a conceptual or physical artifact, in both a forwards and backwards direction (i.e., from its origins, through its development, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases) [24]. The purpose is to help understand an artifact such as a workflow schema, from the beginning, through design, implementation, and maintenance. Gotel and Finkelstein [11] highlight the differences between post and pre-specification traceability. Whereas the former is concerned about where a particular specification came from the later is concerned about how it relates various artifacts are created to satisfy it. In the context of workflow systems, pre-specification traceability may correspond to the rationale behind the workflow schema specifications and post-specification traceability may correspond to workflow instantiations to satisfy a given specification.
Ramesh and Jarke [24] provide a detailed account of the various roles played by traceability in supporting complex organizational processes such as complex systems development. Traceability can be seen as a measure of the quality of the systems development process and can facilitate communication among various stakeholders involved.

The need for traceability has been long recognized in complex design activities such as large scale software development. In fact, standards governing the development of complex, computer-based systems such as the MIL-STD-2167A (DoD, 1988) even mandate traceability. However, recent studies observe that there is wide variation in the quality and usefulness of traceability information among system development efforts. This is due in large part to the absence of clear guidelines, in either the standards or the current literature, on what traceability information must be captured and how it should be used. Recent research also suggests that the lack of pre-specification traceability is another important reason for problems associated with current traceability efforts [11]. Recent research fills this void and has created several reference models that can be used to create context specific traceability schemes to support varied stakeholder needs.

4 DWFM – Need for knowledge management

In this section, we discuss the need for augmenting current workflow management systems with the ability to represent process knowledge about their development and use. We present a snapshot of a case study in a manufacturing industry to highlight critical problems faced in developing and maintaining workflow systems. In the next section we outline our approach to addressing these problems using a traceability based knowledge management solution.

Consider the workflow to automate the manufacturing control process shown in Figure 1. This includes a workflow component that shows the tasks in manufacturing parts. After obtaining orders, the manufacturing management (MM) enquires material controlling (MC) department (1). MC gets BOM (Bill of Material) from DCC (2,3). MC enquires inventory for raw materials (4,5), and reports to MM (6). MM notifies the process-planning task (7). Using the process-plan and raw materials products are produced. Qualified products are sent to the product inventory (11,12,13).

Figure 1: Manufacturing Control Process Workflow

To accommodate changes in business needs, the organization decides to outsource production of some parts. The manufacturing control process workflows need to be modified to accommodate this change. First, a new workflow to handle is created. The outsourcing workflow, shown in Figure 2 represents this revised schema. The manufacturing management (MM) sends orders to outsourcing companies. At the same time, DCC sends the parts requirements, and inventory provides raw materials to them. After the parts are produced, they are sent to product inventory. To accommodate such a change, workflows that need to be redirected in the model specification, or in workflow schema are identified. Finally, dependencies among workflows need to be maintained to avoid inconsistencies as well as to ensure correctness of the changed specifications.
The above scenario highlights a variety of issues that need to be addressed in accommodating such changes. We discuss three critical issues here.

First, as organizations are facing continuous changes, workflows have to be modified correspondingly to reflect the changes in business processes they support. The need for documentation of such changes in workflows has been emphasized in workflow literature [12]. It is argued that the main purpose of documentation in the context of workflow is human understanding. Users need to be able to understand what systems mean in business terms and what requirements do the systems satisfy. The documentation should also be able to suggest the effect of any proposed change. As business-process descriptions are typically mapped to workflow schemas in one big leap [13], interpreting and mapping is often time consuming and very difficult. All the design decisions made during the process are typically not documented and the rich process knowledge behind these mappings is often lost forever. We argue that this mapping must be explicit and fine-grained enough that it is possible to understand where a workflow segment came from.

Second, the lack of appropriate process knowledge may result in not only poor understanding of the workflow processes, but also hinder the ability of the organization to respond quickly to changing environmental conditions. For example, in the absence of detailed mapping between business needs, processes and workflows, it will be difficult or impossible to fully evaluate the repercussions of changes needed in the workflow systems to address changes in the environment. This problem is particularly acute dynamic business contexts where workflows are modified frequently. Often rationales for current processes and workflows have to be re-constructed before changes can be made. This process is prone to error and the potential for creating designs that may conflict with original design policies is very high.

Third, workflow re-configurations may not be reliable and consistent [8]. Re-configuration can happen to either the workflow specifications or executing workflow instance directly. While re-configuring workflows, it is important to identify all the dependencies among workflow fragments that are changed to identify and avoid inconsistencies. Since current workflow management systems have serious limitation in both the workflow model and the workflow execution environment with respect to failure detection [8], facilities to guarantee of workflow correctness can be very valuable. Sometimes, re-configurations are only partially done, leading to serious consequences. For example, in our case study, the cost accounting workflows located in the financial module will be affected by changes made in the manufacturing process. After parts/products are manufactured, and before sent to the temporary inventory, the workflow system will trigger the task that calculates unit cost of production. However, when the manufacturing process control workflow is modified to support outsourcing, maintenance engineers may fail to detect changes necessary in other workflows such as the cost accounting workflow. Then, the unit cost of the product computed by this system will be erroneous.

5 Our Approach

In this section, we propose an approach to managing process knowledge in the development and maintenance of workflow systems and their instantiations using traceability. Traceability provides the ability to cross-reference various physical and conceptual artifacts produced in an organizational process such as workflow specification and execution. Our approach is based on the premise that a variety of tools can be developed to support the specification, maintenance, and instantiation of workflows if a rich history of their development can be captured using a traceability scheme. This raises the question: what are the components that need to be traced in such a history. We have developed a conceptual model.
to address this question. Figure 3 shows the conceptual model that identifies important elements in a traceability scheme to represent the process knowledge in workflow system design, maintenance and implementation. The model is a specialization of reference models for traceability developed in [24] to represent complex organizational processes. The model consists of three major types of objects among which traceability is maintained, viz, Business Process Objects (BP objects), Decision Objects and Workflow System objects (WFS objects). The model also explicitly identifies the stakeholders involved in management (creation, modification and maintenance) of these various kinds of objects.

To build a workflow management system, business process model must be derived from business requirements. The need for supporting business processes lead to a set of requirements that must be satisfied by a workflow system. These requirements may be in the form of constraints (say, organizational procedures and rules that must be followed) as well. Workflow schemas and tasks that they are composed are examples of WFS objects. The model explicitly represents the traceability links between BP objects and WFS objects. For example, the links between a workflow schema or a task to a constraint or a specific requirement that leads to its creation can be explicitly represented using this scheme.

Business process objects and workflow objects can be represented in more detail to support their complete specification. For example, various tasks, dependencies among them, and the order of tasks can be specified in such a scheme. For each task, the stakeholders that are involved in the definition and execution can be specified. Further, triggers for each task may be specified using rules of the form $T = \{E, C, A\}$, where $E$ represents a set of events that can cause the activation of the trigger, $C$ represents the conditions evaluated when active the trigger, and $A$ represents a set of action performed if the events satisfy the conditions defined in $C$.

During the process of definition of workflows, various components get specified, modified, and elaborated. This process may consider many design alternatives. After analyzing and evaluating, some alternatives are rejected. During this process, the rationale and assumptions behind critical decisions can be captured. By linking BP objects, workflow objects and decision objects, the history of each can be easily traced. When change need to be made in any component, the repercussions of changes can be easily ascertained. Further, potential conflicts that may arise due to intended changes could be easily identified.

The framework described here consists of elements to represent the actors, inputs and outputs of the workflow system development process as well as the linkages among them. It can be used to represent the following dimensions of traceability information.
What information is represented - including salient attributes or characteristics of the information?

For example, links among the inputs and outputs of the customization process such as Requirements, Assumptions, Designs, System components, Decisions, Rationale, Alternatives, Critical Success Factors, etc. must be maintained. These represent the major conceptual elements among which traceability is maintained during the various stages of workflow systems development.

Who are the stakeholders that play different roles in the creation, maintenance and use of various conceptual objects and traceability links across them?

Where it is represented - in terms of sources that “document” traceability information.

Examples of sources traceability information include business process objects such as requirement specification documents, meeting minutes, design documents, memoranda, telephone calls as well as references to various stakeholders.

How this information is represented - both by formal and informal means and by how it relates to other components of traceability?

Why a certain conceptual object was created, modified, or evolved?

The rationale behind the creation, modification, and evolution of various conceptual and physical objects need to be maintained. This information may include issues, alternatives, and arguments supporting and opposing various implementation decisions.

When this process-related information was captured, modified, and evolved.

6 Supporting WFKMS with Traceability

We now discuss the use of the above framework in a prototype traceability knowledge management systems created to support WFMS. This system is capable of supporting the instantiation of the various elements in our conceptual model through a graphical interface. The system is intended to be used by various stakeholders involved in the creation and maintenance of workflow schemas and the business processes they support. We describe the functionalities of the system through three short scenarios corresponding to the ones described in section 4. The scenarios will address how the functionalities can alleviate the problems described earlier.
6.1 Traceability to context

First, we describe how the context in which a workflow schema is specified can be linked to it. Figure 4 shows a fragment of traceability knowledge that can be established between requirements and workflow objects. Returning to our example on the definition of workflows for manufacturing process control, observe that this workflow was guided by a requirement to support manufacturing. The requirement to either support manufacturing or support outsourcing are necessary to support the business process related to product acquisition. The option to support manufacturing is based on the assumptions that it helps control quality and maintain the core competencies of the organization. The later also objects to supporting outsourcing. However, outsourcing is supported by the assumption that it may lead to lower costs.

The scenario illustrates that a rich context in which a workflow is defined can be captured at varying levels of detail using our approach. This helps in improving the understanding of the workflow as well as its maintenance.

6.2 Dynamic Re-configuration

An important use of traceability knowledge is in the reconfiguration of workflows with changed business environment and processes. Recall from our case study that the organization may be forced to move from manufacturing to outsourcing due to changes in the environment. In our example, the change in the validity of the assumption that outsourcing the product may result in losing core competencies and the validity of the assumption that it is cheaper to do so may suggest moving to outsourcing rather than manufacturing. The dynamic reconfiguration of the workflows from that shown in Figure 1 to that shown in Figure 2, in this instance, can be supported by our system with just changes in the validity of these assumptions. Our system uses a reason maintenance system to propagate the effects of changes in
one component of process knowledge onto other. In this specific example, the system can suggest that the changed context should involve executing workflows specified in Figure 2.

6.3 Maintaining Integrity

Another important problem identified in our analysis is the difficulty in maintaining the integrity of workflows across organizational and task boundaries. For example, in Figure 5, a change in the manufacturing control process to include outsourcing has implications for workflows in other subsystems such as the cost accounting system in the financial module. Recall that the unit cost computation in this module assumes that the product is produced in-house. However, with the changes in the business process to move to outsourcing this assumption gets invalidated. The make/buy decision made in the manufacturing context, in effects, affects the workflows for computing unit costs. Our system provides the ability to manage such dependencies as well. When the workflows in the manufacturing control process change the system will prompt the user about potential conflicts that could arise. In the event that the repercussions can be formally modeled, the system can also suggest a reconfigured workflow for the cost accounting process.

In the brief scenarios described here, the capabilities of the system to address three key issues associated with workflow specification and maintenance are illustrated.

7 Related Work

Extensive research has been done on dynamic change within workflow systems. Maurer et al. [14] had designed MILOS system to support dynamic coordination of distributed software development teams by integrating project planning and workflow technologies over the Internet. It uses a workflow engine to monitor and implement all changing events. Shrivastava et al [15] proposed the Reliable Workflow Systems (RWS) based on CORBA, which uses task controllers to manage dynamic reconfigurations. Criticizing the RWS’s tightly coupled system structure, Tari et al. [8] proposed a framework based on CORBA using even-channels. Weske [16] studied the ability of WFMS to adapt the structure of running workflow instances to modified workflow schemas.

New research issues come up with the proliferation of studies done on dynamic workflow management systems. Dynamic configuration involves either the work specification or the executing workflow instance. Changes to procedures, performed in an ad-hoc manner, can cause inefficiencies, inconsistencies, and catastrophic breakdowns [7], [8]. Ellis et al. [7] attempted to solve part of this problem by introducing a mathematical formalism to model and analyze dynamic structural changes within workflow systems. Reichert et al. (1997) presented a framework for the support of ad hoc structural changes of WFMSs. Those approaches are based on the knowledge of workflow schemas, rules, and constraints. With the traditional design document, identifying relevant workflow schema and constraints on the flow is a time-consuming job.

Apart from problems posed by structural changes, there are a variety of issues raised by changes that might conflict with the underlying business goals, rules, and regulations implicitly stored in workflow systems. Zhao [17] indicates that different types of knowledge are contained in organizational WFMS. This includes process knowledge that describes tasks, roles, rules, and routes, institutional knowledge that describes business procedures and regulations, and environment knowledge that describes environmental factors such as government. That knowledge is built into WFMSs when designing workflow schema, discussing alternatives, and solving arguments among different end-users. However, some of them are not clearly represented in current workflow system models. Without realizing the constraints, dynamic changes may conflicts with business processes.

The importance of process knowledge and the context in which the process knowledge is created has been well recognized in recent research. Kwan [18] examined several workflow modeling methods and concluded that the current process models aimed at supporting process enactment only support partial views of a process. Based on Curtis’s [19] four perspectives in process models, Kwan [18] defines a process meta-model to represent process knowledge components such as goals and sub goals, and the links among them. Our work is similar in spirit that we provide mechanisms to capture the rich contextual knowledge in which workflow systems are built, maintained and enacted.

Several frameworks are proposed to support the usage of WFMS as the basic infrastructure in the development of organizational knowledge management systems [20], [21]. However no work has been done to study how to manage the processes and institutions’ knowledge implicitly stored in workflow systems, especially in a dynamic environment.

Use of traceability to manage complex organizational processes have been explored in a variety of domains. For example, the importance of requirement traceability has been well documented in the software engineering literature. For example, REMAP system provides facilities to capture and reason with design rationale in domains ranging from concurrent engineering to new product development [22]. Pinheiro et al. describe TOOR (Traceability of Object-Oriented Requirement) system that uses multiple ways to trace requirements to support
maintenance activities. PRIME [23] (Process Integrated Modeling Environments) framework empowers method guidance through process-integrated tools. The framework defines tool models, integrates tool models with method definition, interpret the environment and synchronize the tool and enactment mechanisms thereby supporting comprehensive traceability capture and use.

Ramesh and Jarke [24] present reference models for requirement traceability based on an extensive empirical study. These reference models not only provides a link from initial requirements to the actual system components, but also cover detailed traceability in various systems engineering activities such as requirement management, design allocation model, and compliance verification model.

8 Discussion and Future work

The proposed approach to maintaining traceability requires careful consideration of several issues. First, the overhead involved in capturing detailed history including design rationale, assumptions, decisions, and constraints, can be very high. Second, though traceability can help improve correctness of dynamic re-configurations, a variety of factors such as the formalisms used in defining workflows, the architecture of the WFMS components and the formality with which traceability knowledge is represented will determine the degree to which this process can be automated. Finally, the integration of traceability tools with workflow management systems may also pose interesting challenges.

We are currently developing a traceability knowledge management system to support WFMS. XML will be used to represent ‘documents’ containing traceability knowledge. For example, Xlink [25] will be used to represent links between the elements in the conceptual model. XLink provides a framework for creating both basic unidirectional links and more complex linking structures. It allows XML documents to assert linking relationships among more than two resources, associate metadata with a link, and express links that reside in a location separate from the linked resources. Xlink can not only represent a relationship between two components (XML files), but also can capture knowledge of the link. Our environment is intended to provide a variety of mechanisms to support the various tasks involved in workflow management and maintenance. Further, we are investigating mechanisms for non-intrusive capture of traceability information to reduce the overhead involved.

Finally, the empirical evaluation of the effectiveness of the proposed approach is also a subject of current work.

9 References


