Information Knowledge Systems Management 6 (2007) 61–80
IOS Press

Workflow management for enterprise transformation

James Caverlee[a,b], Joonsoo Bae[c], Qinyi Wu[a], Ling Liu[a], Calton Pu[a] and William B. Rouse[b]
[a]*College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332-0280, USA*
[b]*Tennenbaum Institute, Georgia Institute of Technology, 755 Ferst Drive, NW, Atlanta, GA 30332-0205, USA*
[c]*Department of Industrial and Information Systems Engineering, Chonbuk National University, 664-14 Duckjin-dong Duckjin-gu, Jeonju, Jeonbuk, 561-756, South Korea*

Abstract: Workflow management is a core component of modern enterprise information technology infrastructure that automates the execution of critical business processes. Since enterprise transformation typically introduces changes to the corresponding business processes, it is important for modern workflow management systems to provide effective support for seamless incorporation of these changes. In this paper, we examine a collection of selected workflow concepts and techniques that are significant for dealing with transformational changes. Especially, we will focus on notions and techniques that are directly relevant to enterprise transformation, such as workflow patterns, workflow adaptation, and workflow data mining and merging. We also include a short summary of business process management, the fundamental concepts of workflow management, and a discussion of workflow support for enterprise transformation to keep the paper self-contained.

1. Introduction

As the market landscape changes in an increasingly global inter-connected economy, enterprises face complex challenges of staying relevant to their customer base, maintaining competitive advantages over rivals, and anticipating the growth and needs of an uncertain future. Strategic management of enterprises must increasingly be focused on fundamental changes with respect to markets, products, and services [35]. For enterprises to be successful, they must increasingly move beyond incremental improvements to existing products and services to wholesale transformation.

Supporting enterprise transformation from an information technology (IT) perspective requires effective management of critical changes to existing processes and workflows *and* efficient support for the deployment of new processes and workflows. In this paper, we address these key challenges to supporting the overall goal of enterprise transformation and offer potential solutions from workflow management. In particular, we discuss the main modeling technique for capturing and supporting enterprise transformation – the well-known approach known as *business process management*. We discuss how *business process management* is useful for conceptualizing and modeling enterprise transformation and how *workflow management* provides concrete techniques for implementing and automating business process management. Workflow management is the most widely accepted implementation of business process management and a critical component for successful enterprise transformation. The key challenges to workflow management in this context are managing the efficient and effective automation of workflows and agilely handling the change inherent in enterprise transformation.

We examine a collection of selected significant workflow concepts and techniques that are particularly well-suited to the challenge of enterprise transformation. The first technique – *workflow patterns* – provides a unified way of describing workflows, replicating the logic throughout related business processes, and providing enterprises with the leverage to revise existing workflows and quickly roll-out new workflows. The second technique concerns the *adaptation of workflows* to dynamic environments, like the ones typically encountered during transformative times. Finally, *workflow data mining and merging* provides ample opportunities for enterprises to extract competitive business advantages from their existing workflows and yield new opportunities for fundamental reshaping of enterprise processes.

2. Motivation: Enterprise transformation

In a dynamic competitive landscape with uncertain opportunities and increasing demands on their business processes, enterprises must increasingly be agile with respect to their customer-base, competitors, and the evolving market landscape. Transformation is at the heart of many enterprise decisions and is the subject of an increasing body of literature and of great concern to modern executives. To illustrate what we mean by *transformation*, we highlight several prominent examples of enterprise transformation:

- Prior to eBay, the market for resale consumer items was powered mainly by classified advertisements and off-line person-to-person arrangements. eBay *transformed the market* by building a completely online marketplace that redefined the notion of a resale market.
- UPS fundamentally *transformed its offerings* by expanding from package delivery to supporting the entire supply chain management for its clients. UPS transformed its value proposition to customers via new product and service offerings.
- Starbucks *transformed the perceptions* of coffee drinking by repositioning coffee as an affordable luxury item.
- Nucor Corporation – the largest steel producer in the United States – *transformed its operations* by adopting mini-mills which drastically reduced the costs of steel making relative to traditional approaches. This productivity improvement via extensive process improvements transformed Nucor's value provided to customers, suppliers, and employees and increased shareholder value.

These examples illustrate several flavors of transformation – transformation of the market, of offerings, of perceptions, and of operations. At its core, each type of enterprise transformation is driven by experienced and/or anticipated value deficiencies. Value deficiencies may be characterized as opportunities, threats, competition, or crises:

- Value Opportunities: The lure of greater success via market and/or technology opportunities prompts transformation initiatives
- Value Threats: The danger of anticipated failure due to market and/or technology threats prompts transformation initiatives
- Value Competition: Other players' transformation initiatives prompt recognition that transformation is necessary to continued success
- Value Crises: Steadily declining market performance, cash flow problems, etc. prompt recognition that transformation is necessary to survive

These value deficiencies result in significantly redesigned and/or new work processes as determined by management's decision making abilities, limitations, and inclinations, all in the context of the social networks of management in particular, and the enterprise in general. Of course, there are several critical factors that impact enterprise transformation, including the appropriate allocation of attention and resources, the structure of management decision making, and the social networks that provide the foundation for transformation adoption and implementation.

Fundamental to enterprise transformation is the interplay between the overarching goals of transformation and the fundamental *work processes* that drive an enterprise. An enterprise's approach to work processes powers the transformation. Enterprises may choose to improve how its work is currently performed, change how current work is performed, or perform different work altogether. For example, an enterprise may fundamentally transform its business processes by:

- Targeting New Markets: e.g., pursuing global markets such as emerging markets, or pursuing vertical markets such as aerospace and defense
- Employing New Market Channels: e.g., adding web-based sales of products and services such as automobiles, consumer electronics, and computers
- Changing the Value Proposition: e.g., moving from selling unbundled products and services to providing integrated solutions for information technology management
- Revising its Offerings, e.g., changing the products and services provided, perhaps by private labeling of outsourced products and focusing on support services
- Outsourcing and Offshoring: e.g., contracting out manufacturing, information technology support; employing low-wage, high-skill labor from other countries

In this paper, we focus on work processes that are conducive to automation, especially via workflow management systems. Some examples include:

- Process Standardization: e.g., enterprise-wide standardization of processes for product and process development, R&D, finance, personnel, etc.
- Process Reengineering: e.g., identification, design, and deployment of value-driven processes; identification and elimination of non-value creating activities
- Web-Enabled Processes: e.g., online, self-support systems for customer relationship management, inventory management, etc.
- Supply Chain Restructuring: e.g., simplifying supply chains, negotiating just-in-time relationships, developing collaborative information systems

There are multiple, overlapping opportunities with respect to workflows and enterprise transformation. An enterprise facing transformation will undoubtedly need to revise, re-engineer, or transform its fundamental ways of conducting business – its fundamental processes and workflows. In this case, the enterprise's goal of transformation necessarily requires and drives the transformation of its workflows. Conversely, it may be the case that the enterprise's workflows and the competitive knowledge derived from analysis and mining of these workflows may motivate and facilitate the overall enterprise transformation.

In the rest of this paper, we further discuss process management and how enterprises may successfully leverage existing approaches. We focus our discussion on workflow management techniques – introducing the fundamentals of workflows and discussing several advanced topics that provide support for enterprise transformation.

3. Process and workflow management

To respond to the challenges of transformation, enterprises require proven methods and technologies for effectively and efficiently enabling the transformation. Some of the challenges facing enterprises include identifying value deficiencies that may guide transformative opportunities, deploying the transformation throughout the enterprise via revised business processes and novel processes, and managing the transformation lifecycle.

3.1. Business process management

The main modeling technique for capturing and supporting enterprise information technology infrastructure is the well-known approach called *business process management*. Simply speaking, business process management consists of an explicit description and representation of the coordination, optimization, and automation of the enterprise assets and tasks – whether internal or external – that make up an enterprise's business processes. As an explicit description, it is directly affected by an enterprise adapting its business processes to new organizational needs. As a representation of automated processes, it is an important step in the implementation of new processes towards enterprise transformation.

In general an organization can potentially benefit from adoption of business process management in the following aspects:

- It provides a clear view of business structure, which facilitates the identification of process bottlenecks and the reuse of existing infrastructure.
- It increases output quality by providing uniform and consistent management of information processing over increasingly heterogeneous subsystems.
- It automates the low-level management workload like monitoring work in progress or job scheduling, so enterprises may focus on high-level opportunities.

Of course, enterprise transformation provides some unique challenges that go beyond the traditional scope of process management – meaning effective solutions will draw on existing techniques like business process management as well as drive new opportunities for solutions specifically tailored to enterprise transformation, e.g. to further enhance the capability of an enterprise to react to unanticipated market changes and to anticipate market opportunities for expanding market value.

Business process management has evolved as a significant conceptual modeling tool in the past decade. Three driving factors have pushed its widespread adoption:

- Focus on Processes: Enterprises have realized the necessity for the shift from ad-hoc design to organic growth by decoupling process management from applications. Nowadays, business process management is being recognized as a way to align and optimize departments across the enterprise.
- Technology Advances: The rise of the Internet and advances in human computer interaction have removed two obstacles to business process applications that existed in the 1980s (Aalst et al. 2003a). The backbone of the Internet is widely established throughout the world. Meanwhile, human-computer interface is friendly enough for non-specialists.
- Adoption of Standards: The standardization of communication protocols like the Web services standards combined with workflow technologies have resulted in the adoption of extensible and flexible service-oriented architectures that promote enterprise application integration and business-to-business integration.

To illustrate business process management, we show in Fig. 1 the standard business process management lifecycle [1] that describes the four fundamental phases in support of operational business processes – (1) process design, (2) system configuration, (3) process enactment, and (4) diagnosis. Processes are initially designed and then re-designed in the design phase. Subsequently, these designs are implemented in a workflow management system (or similar system) in the configuration phase. The

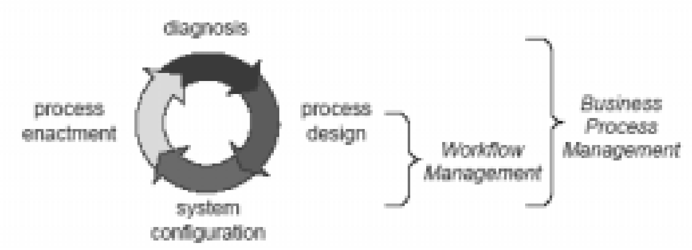

Fig. 1. Business process management lifecycle.

enactment phase consists of the actual execution of configured business processes using the workflow management system. Finally, processes may be analyzed in the diagnosis phase to identify process deficiencies and other areas of improvement [2].

3.2. *Workflow management*

Figure 1 displays the relationship between business process management and *workflow management*. Workflow management is the core IT component for implementing and executing the processes and tasks defined by business process management. Generally, the term *workflow* is used in a technical context to describe an ongoing series of connected interactions between people and an enterprise's information technology systems. In contrast, business process management has a broader meaning; it typically encompasses non-technical issues surrounding the design and analysis of processes and the impact of these processes on the enterprise, as in Fig. 1.

There is wide industry support for workflow management, and a growing number of vendors offer workflow management solutions, including TIBCO, IBM, and Microsoft. In addition, several industry bodies – like the World Wide Web Consortium, the Workflow Management Coalition, the Business Process Management Initiative, and the Organization for the Advancement of Structured Information Standards – are publishing workflow standards that promise increasing corporate adoption of workflow management systems.

While workflow management systems are very useful for daily operations, their role becomes critical during enterprise transformation. As business processes change and adapt, workflow management systems must make corresponding changes and execute adaptation actions to implement the enterprise transformation correctly, efficiently, and reliably. We refer the interested reader to a number of studies that describe workflow management in enterprise settings, including telecommunications [37], procurement and supply chain management [6], and agricultural zoning [17].

Introducing workflow management in an enterprise delivers benefits both for the development of enterprise software applications as well as for the overall business goals. A workflow management system eases the costs of developing new enterprise applications and reduces the maintenance of standard enterprise software. According to Baeyens [9], enterprise application costs are reduced due to:

- Lower application development risk: By relying on a workflow management system, the enterprise application developer may avoid the traditional error-prone approach of translating from user requirements to software design. Instead, the developer and the business analyst will have a common understanding of the appropriate process states and actions.
- Centralized implementation: By providing a unified perspective to an enterprise's processes, a workflow system provides an advantage versus a traditional implementation consisting of a combination of software fragments scattered over various systems.

– Rapid application development: Since the workflow management system provides support for tracking the many participants in a process, the enterprise software need not be burdened with these administrative tasks, leading to faster application development and code that is more easily maintained.

With respect to the overall business goals of an enterprise, a workflow management system supports the formal description and analysis of business processes, resulting in a number of significant gains, including:

– Process improvement: The formal description and analysis of business processes allows an enterprise to identify inefficiencies in existing processes, leading to streamlined and simpler processes. Similarly, the adoption of standard workflows and the availability of audit trails improve the overall quality of processes.
– Enterprise flexibility: Since the workflow management system automates so many of the administrative tasks and provides a unified perspective on processes, enterprises have greater ability to adjust existing processes and to deploy new processes as the business needs change.

From the IT point of view, two important reasons for enterprises to adopt a workflow management system are: ease of software development and reduction of risk for overall system development. Using the high level programming language constructs of workflow specification languages is faster and less error-prone than writing custom software. Consequently, a workflow management system simplifies the deployment of new versions of business processes. Furthermore, since workflow management systems are relatively mature products, a well-defined workflow produced by business process analysis usually works as advertised for existing processes. The question addressed by the following section is the workflow support of process changes during enterprise transformation.

4. Selected workflow management techniques

As we have seen, the challenges of enterprise transformation create new burdens on the management, evolution, and wholesale rethinking of traditional business processes. And the presence of workflow management systems provides enterprises with an opportunity to better manage their transformation. In this section, we focus our discussion of workflow management in direct support of enterprise transformation. To make this section self-contained, we include an overview of the reference workflow architecture, the key modeling concepts of workflow management, and popular workflow specification languages. In the following three sections, we discuss three relevant examples of workflow automation techniques that are particularly well-suited to the challenge of enterprise transformation. First, we describe *workflow patterns,* which provide a unified way of describing workflows, replicating the logic throughout related business processes, and providing enterprises with the leverage to revise existing workflows and quickly roll-out new workflows. Then, we describe *adaptation of workflows* to dynamic environments, like the ones typically encountered during transformative times. Finally, we outline *workflow data mining and merging* techniques that provide ample opportunities for enterprises to extract competitive business advantages from their existing workflows and yield new opportunities for fundamental reshaping of enterprise processes.

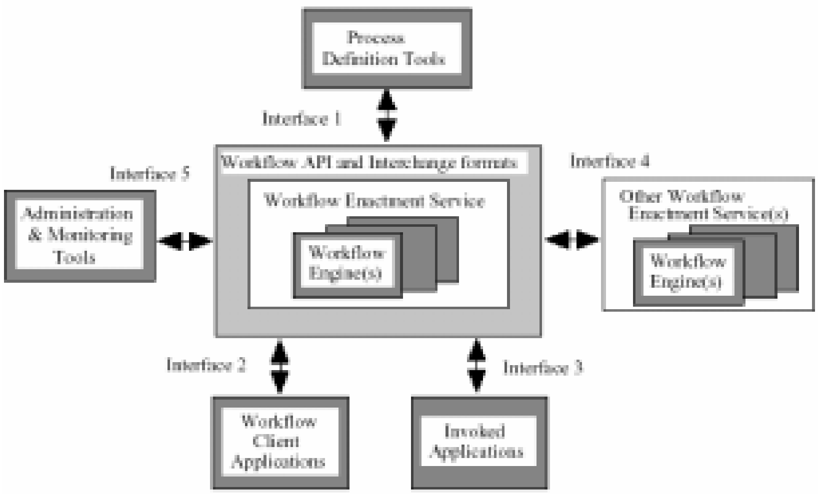

Fig. 2. Reference workflow architecture.

4.1. Workflow Basics

4.1.1. Workflow reference model

The Workflow Management Coalition defines a workflow system as "*a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications*" [36]. Figure 2 shows the reference architecture recommended by the coalition. It consists of an engine, a process definition tool, workflow application clients, invoked applications, and administration and monitoring tools.

The *workflow engine* and its associated tools manage the workflow process control data (which is used by the workflow management system only) and the workflow process-specific data (which is used by both applications and the workflow management system) which is typically stored in a workflow database. The type of data stored, accessed, and updated by the workflow engine typically includes statistical information about existing processes, control information necessary for executing and monitoring active process instances, and generic workflow activity schemas.

The *process definition tool* defines the possible paths through the business process model – both those paths that are explicitly defined and those that may be inferred from the process model. Additionally, the process definition tool identifies relevant rules that govern the process paths and the associated actions that need to be executed.

The *monitoring tools* allow the workflow administrator to monitor past workflow process activity status and current workflow process execution. Each workflow instance may be viewed to understand the priorities, deadlines, and significant participants. Enterprise managers may rely on real-time status reports for evaluating employees, understanding the state of active workflow instances, and assessing the overall performance of the processes.

The *simulation tools* support the verification of workflow process behavior by simulating actual workflow instances with sample data. Simulation is especially significant for analyzing complex workflows with non-obvious process evolution.

The *analysis tools* support advanced analysis of workflows beyond the simple monitoring provided by the monitoring tools. Typically, the analysis tools may identify process logic errors and inefficiencies in the overall process system by evaluating audit logs that maintain the history of the status of various workflow system components. The quality of the analysis tool can have a profound impact on the development and deployment costs of processes and the quality of running processes.

4.1.2. Workflow model concepts

Given the workflow reference architecture, we next focus on the key concepts necessary for workflow modeling.

A *process* is a template specifying a set of tasks and their execution dependencies. The process template is intended to be an abstract pattern for specifying business logic. Processes may be composed of subprocesses, which again consist of activities and dependencies, and possibly even further subprocesses.

A *process case* is an instance of a process template. Each case has a unique identity used to refer to the case in question. Its execution follows a particular execution path prescribed by the process template. It contains the actual data, external events and consumes resources.

A *task* is the basic unit of work of which all process cases are composed. A task is meant to be a conceptual unit of work; the actual execution of a task typically requires the execution of a sequence of *activities* (or sub-tasks). Each activity may consist of sequence of interactions between the enterprise software, the human agents responsible for the task, and the workflow management system. If a task fails or results in a system bottleneck, the workflow system may choose to roll back the task and execute an alternative task.

Process enactment is the determination of a particular execution path of a process case. Typically, the execution possibilities for a process case are defined during the design phase of the workflow. Example execution possibilities include sequential routing (in which tasks are executed in simple serial fashion), parallel routing (in which multiple tasks may execute concurrently), selective routing (in which the workflow system makes a choice between two or more tasks), and iterative routing (in which a task is executed more than once for a particular process case). We illustrate some of these routing patterns in the following section.

A *resource* is an entity that can carry out a particular task. Resources could be human agents of the enterprise, an enterprise software application, or any other system capable of contributing to the execution of a task. A group of resources is called a *resource class*. If the grouping is functionally based, it is called a role, like grouping together secretaries, faculty, and system administrators.

Resource management refers to the allocation of resources to specific tasks. The process definition tool defines the appropriate paths of a business process and the tasks necessary for completion, but it does not indicate which resources should be responsible for each task. Resource allocation is critical to ensure the efficient and effective completion of processes [1]. There are various resource management strategies like first-in first-out, shortest processing time, earliest due date, and so on [20].

4.1.3. Workflow specification languages

There exist a number of languages for specifying workflows: procedural, declarative, rule-based, data-flow-based, visual, and Petri-net, among others. Most workflow specification languages provide support for the declaration of the control flow and information flow through a workflow process, including the synchronization order among the various participants in the workflow. The following section gives a detailed explanation about workflow patterns for abstracting common control flow across several workflow instances. Additionally, state-of-the-art languages support the specification of priorities for scheduling tasks, as well as exception handling for cases in which a task fails or cannot complete.

A workflow management system normally provides support for several specification languages to provide views at different abstraction levels. High-level abstraction helps to define behavior of process

Table 1
Workflow pattern types

Pattern Categories	Pattern Types
Basic Control Flow Patterns	1) Sequence 2) Parallel Split 3) Synchronization 4) Exclusive Choice 5) Simple Merge
Advanced Branching and Synchronization Patterns	6) Multi-Choice 7) Synchronizing Merge 8) Multi-merge 9) Discriminator
Structural Patterns	10) Arbitrary Cycles 11) Implicit Termination
Patterns involving Multiple Instances	12) Multiple Instances Without Synchronization 13) Multiple Instances With a Priori Design Time Knowledge 14) Multiple Instances With a Priori Runtime Knowledge 15) Multiple Instances Without a Priori Runtime Knowledge
State-based Patterns	16) Deferred Choice 17) Interleaved Parallel Routing 18) Milestone
Cancellation Patterns	19) Cancel Activity 20) Cancel Case

participants such as people, software, and sub-systems from an organizational point of view of the process without concerning ad-hoc technologies and concrete implementation details [29]. The lower-level abstraction is used to capture the detailed support for the process [30,33]. It includes the protocols between different participants, event routing, and tools. Most workflow management systems normally automate the mapping between the high-level abstractions to low-level abstraction, allowing workflow authors to concentrate on process modeling.

4.2. *Workflow patterns*

Workflow patterns are derived from the experience of pattern design in the context of object-oriented systems. Design patterns were first catalogued by Gamma et al. [18] to provide a common repository of generic software development patterns that could be applied across a wide landscape of different application domains and through various underlying software implementations. Similarly, we may define *workflow patterns* that allow workflow specifications to be abstracted into re-usable patterns. These patterns may be cataloged for future use and deployed more easily than workflows developed from the bottom-up.

A number of different classes of workflow patterns may be defined based on how the underlying workflow specifications are understood. For example, workflows may be modeled from the point of view of the control flow perspective, the data flow perspective, or the resource perspective. Due to their widespread adoption and importance to real workflow systems, we focus on workflow patterns based on the control flow perspective in the rest of this section, beginning with some basic patterns before considering several more advanced workflow patterns.

4.2.1. *Basic workflow patterns*

Workflow patterns are an ideal opportunity for defining generic workflow constructors that may be easily adapted to new circumstances and deployed into operational systems. Workflow patterns describe how simple atomic activities may be combined to form more complex activities for completing a particular task. In this section, we describe two basic workflow patterns – *sequential routing* and *parallel routing*, as illustrated in Table 1 [3]. These control flow patterns provide reusable components for describing activities and their execution ordering.

The most basic workflow pattern is sequential routing. Sequential routing is defined as a "segment of a process instance under enactment by a workflow management system, in which several activities are executed in sequence under a single thread of execution" [36]. Sequential routing is often deployed in simpler workflow applications or as a sub-part of more complex workflows which will often intersperse sequential routing components with more advanced parallel routing components.

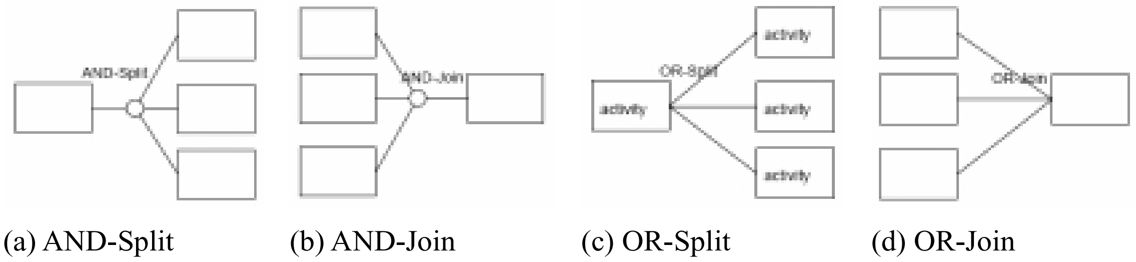

Fig. 3. Parallel routing patterns.

Parallel routing, is defined as "a segment of a process instance under enactment by a workflow management system where two or more activity instances are executing in parallel within the workflow, giving rise to multiple threads of control" [36]. Parallel routing may be further subdivided into four parallel routing patterns – AND-Split, AND-Join, OR-Split, and OR-Join. These four patterns are illustrated in Fig. 3. The AND-Split pattern splits a single thread of control flow in a workflow into parallel threads that may be executed simultaneously. As a result, several activities may be completed at the same time, rather than relying on the simple sequential routing paradigm discussed above. The natural complement to the AND-Split pattern is the AND-Join pattern, in which multiple threads of control are *joined* together into a single thread of control. Since the multiple threads have run in parallel, the AND-Join requires the multiple threads to synchronize before continuing the workflow.

In contrast to the AND-Split is the OR-Split, in which a single thread of control chooses one or more of the several possible branching options (instead of splitting the thread of control into all possible branches as the AND-Split does). The OR-Split is also known as conditional routing, since the branch or branches selected will often depend on the state of the workflow and any conditions that may have been met. The OR-Join pattern joins together multiple threads into a single thread. Unlike the AND-Join, the OR-Join makes no synchronization requirement on the workflow continuing – since no parallel activity execution has occurred at the join point, no synchronization is required.

4.2.2. Advanced workflow patterns

The basic workflow patterns are complemented by five additional categories, as illustrated in Table 1 [3]. These five additional types of workflow patterns provide great opportunity for reuse in real-world workflow applications. The first advanced category provides advanced branching and synchronization patterns beyond those found in the simple sequential and parallel branching operators. These advanced patterns are inspired by actual process scenarios and provide ample opportunity for reuse. The second advanced category – Structural Patterns – imposes restrictions on the workflow patterns. One example is the handling of arbitrary cycles in the workflow. There are a number of advanced patterns for dealing with multiple instances of an activity. These Patterns Involving Multiple Instances provide the ability to synchronize across multiple instances. For example, a journal paper review process may require multiple reviewers to perform the activity Review – only after all instances of the Review activity are synchronized may the workflow proceed. State-based Patterns allow for the incorporation of workflow state into the process, so that choices in the workflow enactment may be made based on the relevant business data available. The final advanced workflow pattern – Cancellation Patterns – allows one activity to cancel another activity. For example, it may be necessary to disable a running HireNewEmployee activity if a job candidate withdraws her application.

These efforts to define workflow patterns yield advantages both for practical workflow deployment and for workflow researchers. Contemporary workflow management systems may provide an insight into the shortcomings of existing patterns, illuminate the relative strengths and weaknesses of current workflow technology, and provide the basis for future developments from the set of existing workflow patterns. With respect to ongoing workflow pattern research, there have been a number of advances using patterns as an infrastructure for resource allocation [23], active rule development [5], exception handling [11,12] and transaction management [19], among others.

4.3. Workflow adaptation

4.3.1. Workflow adaptation process

In enterprise transformation, an important element for success is the degree that the current way of doing business can be quickly changed to a new business mode with the least negative impact on existing infrastructure. Possible adaptations of existing workflows include issues of process redesign, resource reallocation, and the establishment of new business agreements between various parties.

For example, in the context of online marketplaces, businesses seek to provide reduced supply chain overhead, competitive delivery costs, and flexible transaction values to differentiate their products and service offerings in the market. What this means, however, is additional requirements on the systems and tools that support these flexible business practices. An enterprise's transaction system might be required to support multiple payment schemes, like credit card or digital cash. Or, the merchant software might be required to support multiple pay-deliver relationships, like payment before delivery. Beyond these, the fact that it's usually difficult to foresee all the functionalities or services makes the requirements on system support even more demanding. Of course, unforeseen circumstances place a burden on the enterprise and its workflows. A workflow management system is specifically designed to provide flexible support for adapting to unforeseen circumstances without the significant costs typically associated with traditional information systems.

Compared to a traditional information system, an important requirement for workflow management systems is the ability to respond to new changes and dynamically adapt to new requirements. There are two classical paradigms for workflow adaptation: workflow design evolution and dynamic workflow adaptation. The former includes the workflow re-design to reflect those new changes in the business processes of the enterprise due to transformation needs and is typically applied when the need for revisiting the workflow schema arises during or after an enterprise transformation. The latter refers to two types of dynamic adaptations – one is runtime adaptation through compile time design of exception handlers and another is runtime adaptation through dynamic restructuring of the execution schedule of workflow activities while preserving the workflow execution dependency graph [26]. Concretely, the researchers promote the use of exception handlers to explore the modification of normal procedure to accommodate exceptions as they occur [14–16,24]. This approach works extremely well for those exception situations that are well understood and predictable. Alternatively, some researchers promote dynamic restructuring of workflows at runtime through activity split, activity join, and activity unnesting according to a workflow activity precedence graph [25–27]. The main motivation of this approach is to provide exception handling through a set of runtime workflow restructuring operators capable of generating adaptation code at runtime to introduce the flexibility of changing workflow execution scheduling through runtime adaptation. A key motivation behind this approach is the observation that the workflow schema design is valid but some executions of workflow instances may at times require certain scheduling changes to meet the deadlines in the presence of sudden resource shortage or other unexpected delays occurred, especially in a long running workflow of complex activity structure. There is also some research on adaptation techniques that do not require well-defined workflow specification

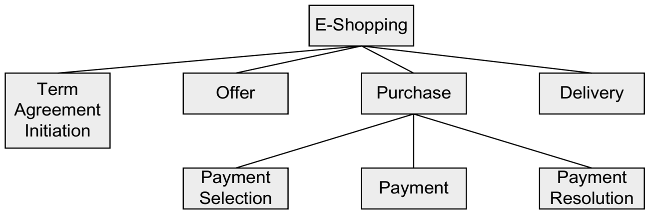

Fig. 4. The online shopping process.

at the design stage, instead relying on partially specified process models and the power of workflow enactment engines to dynamically refine them as needed [10,13,22,28,32].

Workflow adaptation has a number of advantages in terms of value-added flexibility and robustness for management of evolving business processes. There are a number of enabling technologies that drive the advance of workflow adaptation techniques. First, by providing a clear definition of the interaction behaviors between tasks in a workflow, it is easier to compose, exchange, or modify them to form new adapted workflows. Second, there are several techniques from the research community for supporting static and dynamic workflow adaptation while preserving the correctness of business rules. Third, supporting tools like the simulation and monitoring tools mentioned in the workflow reference architecture provide effective assistance in the workflow adaptation process.

4.3.2. Workflow adaptation example

We begin by describing a typical scenario in electronic commerce – shopping over the Internet [38]. Figure 4 illustrates an online shopping process for the E-Shopping activity involving a customer and on online merchant. The task of E-Shopping consists of four sub-activities: TermAgreementInitiation, Offer, **Purchase**, and Delivery. The E-Shopping starts when the customer visits the online merchant's website and enters her login identification to the system. The activity TermAgreementInitiation is first launched, in which the merchant outlines the terms and options. If the customer agrees to the terms set forth by the merchant, then the offer activity follows. After the customer has decided on the goods to be purchased, the E-Shopping process proceeds to the next phase. Depending on the terms both parties have agreed to in the TermAgreementInitiation activity, there are different possibilities for how the two activities Purchase and Delivery are carried out. The activity Purchase has a number of subtasks: (1) PaymentSelection: in which all the necessary payment methods are determined, e.g., whether some coupons or vouchers or frequent-flyer points can be used/redeemed against the purchase, or whether the customer uses credit card or digital cash to pay; (2) Payment: the actual payment process in which value is transferred from the customer to the merchant; and (3) PaymentResolution: in which a receipt is generated by the merchant and given to the customer.

Now let us consider a particular scenario of the E-Shopping activity execution to illustrate workflow adaptation. Suppose a customer has entered an online video store, accepted the terms and agreements, selected two videos that she wishes to watch online, and has just paid using her smart card. Now the video store is about to live-feed the video stream to her. However, the store detects a technical problem that may hinder the delivery of the video stream (e.g. the server may be overloaded due to heavy usage). In order to alleviate the problem, the system suggests to the customer to overlap her delivery with other customers and use multiple deliveries to fulfill the store's obligation. After the user has agreed to this change, the Delivery activity is split into two new activities Delivery1 and Delivery2, each responsible for delivering one episode of the video the customer has ordered. An additional activity dependency is enforced on these two new activities that for the overall Delivery to succeed, both of the new activities

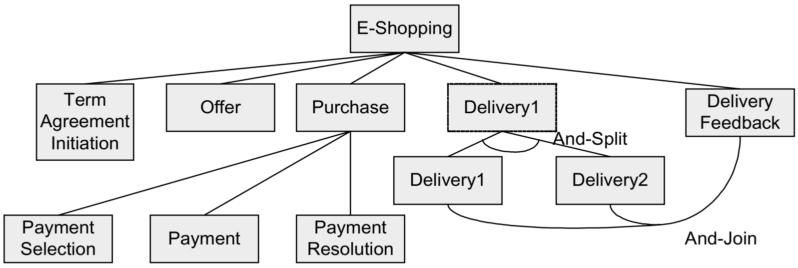

Fig. 5. Process restructuring.

Delivery1 and Delivery2 must succeed. If either fails, then Delivery also fails. The corresponding compensation activity should reimburse the customer's account for the failed delivery. Following this scenario, in order to make sure that customer is not dissatisfied, the store follows up the previous split delivery method with an email right after the delivery service, apologizing for the inconvenience and asking for the customer's feedback. This email initiates a new activity DeliveryFeedback, which is joined with previous two Delivery activities to become a new subactivity of E-Shopping. In case the customer is not satisfied, the store's system may have to compensate the purchase activity with corresponding actions. These kinds of activity dependencies are added into E-Shopping's specifications. The adapted workflow is illustrated in Fig. 5.

This example has relied on a notion of activity adaptation for supporting dynamic business structures that has been introduced in the Transaction Activity Model [39] for dynamic restructuring of ongoing activities. The activity reconstruction operators, Activity-Split and Activity-Join, may improve the execution performance in anticipation of various uncertainties. These operations can help an electronic commerce system bypass temporary or permanent bottlenecks in both data flow and control flow, like those caused by node or network unavailability, user intervention or absence. Furthermore, these two restructuring operators build on a formal notion of validity so that they guarantee a rich set of correctness criteria of the resulting activities. By allowing the release of early-committed resources or ownership transfer of uncommitted resources, "these dynamic activity-restructuring operations highlight a number of advantages, such as adaptive recovery, added concurrency, dynamic process evolution, and enhanced cooperation" [39].

4.3.3. Workflows and business process reengineering

Workflow adaptation normally refers to those changes that happen on a portion of a whole process or extend an existing process with additional functionality. When the process has to undergo fundamental changes, it is called *business process reengineering*.

The goal of business process reengineering is to bring about entirely new business processes that enable drastic improvements with respect to costs, quality, and service. It has been described as "fundamental rethinking and radical redesign of business processes to achieve the dramatic improvements in critical, contemporary measures of performance" [21]. The capability of workflow management systems in process adaptation makes it an essential component in process reengineering. For process redesign in the context of wholesale reengineering, there are several guidelines [2]:

- First the process objectives should be clearly declared to reduce the risk of substantial revision near the late stage of development. The scope of tasks and the level of specialization and generalization should be carefully considered because they have a significant impact on the flexibility and efficiency of the system. One principle to keep in mind is to strive for the simplest possible process to avoid unmanageable processes, e.g. by limiting the number of causal links between different processes.

– Another concern is related to resource management. The process design should be relatively independent of the existence of resources. This can help designers not be distracted by old organization structure. In terms of resource allocation, there are some general roles to follow, like allowing a resource to perform similar tasks in a row and allowing the same resource to work as much as possible on the same case.

Business process reengineering normally requires a thorough analysis. There are various analytical techniques available for establishing these qualitative and quantitative performance indicators using a modeled workflow like Petri-nets [33] and π-calculus [31].

4.4. Workflow data mining and merging

The final workflow management technique we consider with respect to enterprise transformation is workflow data mining and merging. As the importance of process-centric management increases, an organization may leverage its repository of business processes as knowledge assets for exploitation, e.g. by reducing the time and cost of new processes by referencing existing similar processes during new process design and process improvement. Hence it is necessary to identify appropriate process definitions from the already executed process history by understanding the characteristics of the process itself. As a result, there has been an increasing interest in applying techniques from data mining and machine learning to make use of workflow data that are collected during workflow execution. This approach is referred to as process or workflow mining [6,8].

Returning to the business process management lifecycle in Fig. 1, we may see how cases (or workflow instances) may be analyzed in the diagnosis phase once they have been designed and configured. The workflow management system may monitor the running workflow enactments to collect relevant statistical information and workflow logs for use in the diagnosis phase. By analyzing this data using workflow mining techniques, the workflow analyst may discover inefficiencies in the current workflow design, opportunities for overlap between various workflows, and other critical workflow information. Of course, the analysis of these workflows may lead to changes to the workflow during the iterative re-design phase of the workflow lifecycle.

One of the prominent types of workflow mining supported by contemporary workflow management systems is focused on the analysis of workflow audit logs to automatically generate the structured process description underlying the audit trail. This type of workflow mining is especially important when the workflow analyst wishes to extract a workflow design from the set of real executions, rather than explicitly craft the design from scratch. The main idea is to construct a process specification from the workflow logs. These specifications can then be used for diagnosis or redesign of existing processes. There are some basic assumptions necessary for this type of workflow mining – generally, that each event in the log refers to a well-defined task in the workflow corresponding to a specific workflow instance, and that these events are ordered in the log for understanding the relative timing of tasks [4].

For example, in Table 2 [4] we show a simple example workflow log. The log contains information about five workflow instances – 1, 2, 3, 4, and 5. In practice, a workflow log may contain information about hundreds of thousands of instances, each consisting of dozens to hundreds of tasks. In the table, we can see that for four instances – 1, 2, 3, and 4 – the log reveals that tasks A, B, C, and D have been executed. The fifth instance has only executed tasks A, E, and D. Based on this workflow log, we would like to extract the structure of the underlying workflow to understand the relative ordering of tasks and the relationship among the tasks. The workflow miner may begin by observing that all cases begin with the execution of task A and end with the execution of task D. Hence in Fig. 6 we can begin to illustrate

Table 2
Workflow log example

Case Identifier	Task Identifier
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task A
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task E
case 5	task D
case 4	task D

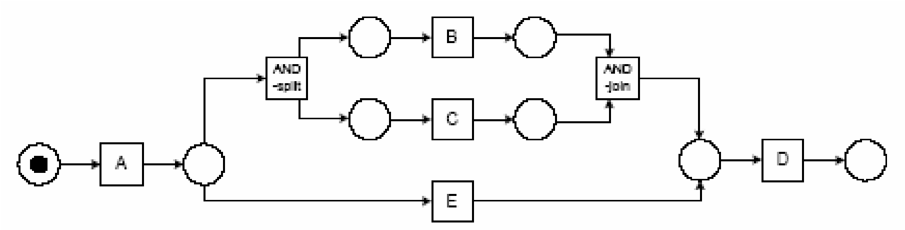

Fig. 6. A workflow model generated from the workflow log.

the underlying process model with starting and ending nodes for tasks A and D. Additionally, we may note that whenever task B is executed then task C is also executed, though the reverse does not always hold. Continuing along this line of analysis, the underlying process model may be constructed. Of course, in practice, the process model may be only partially reconstructed depending on the presence of certain operators in the process and the assumptions made about the audit log.

Alternatively, we may be interested in identifying similar workflows that are necessary to workflow mining and workflow merging [7]. The characteristics of a process can be divided largely into two categories, one is the semantic information that the process has included and the other is information of the process structure itself. The semantic information may include domain information, ontology, work breakdown structure, actors, relevant data, etc. The process structure information mainly includes items such as the components of the process, activity information, how the components are connected, and transition information.

For example, suppose two companies with overlapping capabilities merge to form a new enterprise. In the merged enterprise, a natural challenge is the identification of related workflows for eliminating redundancies, generating new workflow opportunities, and so on. In Fig. 7, we show the case of a target workflow G_0 and five candidate workflows G_1, G_2, G_3, G_4, and G_5. A natural data mining task is to identify the candidate workflows that are most similar to the target workflow G_0. Of course, none of the candidate workflows match G_0 exactly, so the data miner must rely on a measure of similarity.

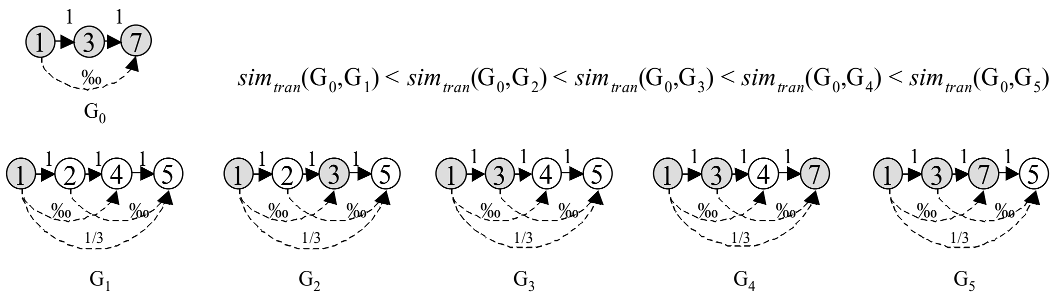

Fig. 7. Transition similarity with inferential transitions.

$$sim_{tran}(G_0,G_1) = \frac{t_0 \, t_1}{|t_0||t_1|} = \frac{0}{\sqrt{2+\frac{1}{4}}\sqrt{3+\frac{2}{4}+\frac{1}{9}}} = 0$$

$$sim_{tran}(G_0,G_2) = \frac{t_0 \, t_2}{|t_0||t_2|} = \frac{1\times\frac{1}{2}}{\sqrt{2+\frac{1}{4}}\sqrt{3+\frac{2}{4}+\frac{1}{9}}} = 0.147$$

$$sim_{tran}(G_0,G_3) = \frac{1\times 1}{1.5\times 1.9} = 0.294$$

$$sim_{tran}(G_0,G_4) = \frac{1\times 1+1\times\frac{1}{2}+\frac{1}{2}\times\frac{1}{3}}{1.5\times 1.9} = 0.490$$

$$sim_{tran}(G_0,G_5) = \frac{1\times 1+1\times 1+\frac{1}{2}\times\frac{1}{2}}{1.5\times 1.9} = 0.662$$

Fig. 8. Example similarity calculation for workflow mining.

In this case we can measure the similarity as a function of the number of common explicit transitions and the number of common inferential transitions. A common explicit transition is a direct transition between common states in both processes. For example, both G_0 and G_3 share one explicit transition from state 1 to state 3. In contrast, G_2 shares one inferential transition with G_0 since the transition from state 1 to state 3 is through an intermediate state 2. Extending this analysis, we can see that G_1 has no common dependency of activities with G_0, G_4 shares one explicit transition and one inferential transition, and G_5 shares two explicit transitions. In Fig. 8, we illustrate an example measure of transition similarity that discriminates among these different cases by considering implicit dependencies as well as explicit dependencies among activities. The similarity measure assigns explicit transitions a value of 1 and inferential transitions a fractional value (e.g. an inferential transition through one-hop is worth $1/2$, through two-hops 1/3, and so on as illustrated in Fig. 7). The similarity measure between two graphs is the product of the shared transitions divided by the product of the size of each graph (in terms of transition values). To illustrate, since G_0 and G_2 share one inferential transition from state 1 to state 3, the numerator is $1 \times 1/2$. In terms of the transition values, the size of G_0 is $\sqrt{1^2 + 1^2 + \left(\frac{1}{2}\right)^2}$. Similarly, we can find the size of G_2 (and of all the other graphs) and calculate the associated similarity. The results of the calculation are shown in Fig. 8, where we see a strict ordering may be made over the candidate workflows. Of course, other similarity measures may be considered, but this example provides some intuition behind the mining of similar workflows.

The topic of workflow mining is related to other management trends such as business process reengineering, business intelligence, business process analysis, continuous process improvement, and knowledge management. Additionally, workflow mining may be used as an input for business process reengineering and continuous process improvement activities.

5. Conclusion and new opportunities

Supporting enterprise transformation from an IT perspective requires effective management of critical changes to existing processes and workflows *and* efficient support for the deployment of new processes and workflows. In this paper, we have addressed several key challenges to supporting the overall goal of enterprise transformation and discussed some potential solutions from the perspective of workflow management. We have discussed business process management as the main modeling tool for enterprise transformation, and described a selection of workflow management technologies that are the key implementation of adaptive business process management.

Concretely, we have examined important workflow concepts and some representative techniques that are particularly well-suited to the challenge of enterprise transformation, such as workflow patterns, workflow adaptation, and workflow mining and merging. The first technique – *workflow patterns* – provides a unified way of describing workflows, replicating the logic throughout related business processes, and providing enterprises with the leverage to revise existing workflows and quickly roll-out new workflows. The second technique concerns the *adaptation of workflows* to dynamic environments, like the ones typically encountered during enterprise transformation. Finally, we have discussed techniques for *workflow data mining and merging* and the ample opportunities of applying these techniques to enable enterprises to understand and improve their competitive business advantages from their existing workflows and discover new opportunities for continued evolution of enterprise processes to ensure their competitive edge and ability of meeting new business challenges.

Of course there are many other important issues relevant to process and workflow management that we have not discussed in sufficient detail here, but are worthy of more attention, including:

- Process Modeling and Design: What is the appropriate process modeling language (e.g., process flow, object-oriented) and what is the appropriate level of granularity for process design? These choices have strong implications on the effectiveness and efficiency of workflow management.
- Workflow Management Deployment: Deploying and implementing workflow management systems in the enterprise poses a number of challenges, including transitioning legacy systems and managing data integration across the enterprise.
- Managing Cultural Change: Workflow management in the context of enterprise transformation places demands not only on the technical infrastructure, but also on the supporting culture of the enterprise. Managing executive buy-in and organizational culture is critical to successful transformation.

We believe that workflow management has an opportunity to fully adapt to the challenges of a world in which enterprise transformation is not only a widely acknowledged need but also a reality that we have to live with. We anticipate the development of new workflow modeling techniques and new workflow management tools that directly address the specific challenges of enterprise transformation. One of the critical challenges for workflow management in the context of enterprise transformation is the natural inclusion of knowledge work into the workflow management system. As enterprises increasingly become havens of knowledge workers, there is a great opportunity for new workflow modeling techniques to capture the knowledge creation, dissemination, and the leveraging of the enterprise's knowledge-base. An additional challenge to workflow management is the natural incorporation of uncertainties into the workflow management system. Enterprises face daily challenges in assessing uncertain outcomes and managing processes with potentially uncertain side-effects. Understanding and analyzing these uncertainties is especially important in the context of "bet-the-company"-style decisions about enterprise transformation.

References

[1] W.M.P. Aalst van der and K. van Hee, *Workflow Management: Models, Methods, and Systems*, 1st edition, The MIT Press, January 1, 2002.

[2] W.M.P. van der Aalst, A.H.M. ter Hofstede and M. Weske, Business Process Management: A Survey, *Lecture Notes in Computer Science* **2678** (2003), 1–12.

[3] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski and A.P. Barros, Workflow Patterns, *Distributed and Parallel Databases,* **14**(3) (July 2003), 5–51.

[4] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm and A.J.M.N. Weijters, Workflow Mining: A Survey of Issues and Approaches, *Data and Knowledge Engineering* **47**(2) (2003), 237–267.

[5] J. Bae, H. Bae, S. Kang and Y. Kim, Automatic Control of Workflow Processes Using ECA Rules, *IEEE Transactions on Knowledge and Data Engineering* **16**(8) (Aug. 2004), 1010–1024.

[6] J. Bae, J. Caverlee, L. Liu, W.B. Rouse and H. Yan, *Process Mining by Measuring Process Block Similarity,* Proceedings of the Workshop on Business Process Intelligence, Vienna, Austria, September 5–7, 2006.

[7] J. Bae and J. Jung, Workflow Clustering Method based on Structural Similarity. Technical Report. 2005.

[8] J. Bae, L. Liu, J. Caverlee and W.B. Rouse, *Process Mining, Discovery and Integration using Distance Measures,* Proceedings of the IEEE International Conference on Web Services, Chicago, IL, September 18–22, 2006.

[9] T. Baeyens, The State of Workflow, http://www.theserverside.com/articles/content/Workflow/article.html, May 2004.

[10] C. Beckstein and J. Klausner, A meta level architecture for workflow management, *Journal of Integrated Design and Process Science* **3**(1) (1999), 15–26.

[11] F. Casati, S. Castano, M. Fugini, I. Mirbel and B. Pernici, Using Patterns to Design Rules in Workflows, *IEEE Transactions on Software Engineering* **26**(8) (Aug. 2000), 760–785.

[12] F. Casati, S. Ceri, B. Pernici and G. Pozzi, Workflow Evolution, *Data & Knowledge Engineering* **24**(3) (January 1998), 211–238.

[13] S.M. Deen, *Cooperating agents for holonic manufacturing,* Proceedings of the Multi-Agent-Systems and Applications, 2001.

[14] M. Divitini and C. Simone, Supporting Different Dimensions of Adaptability in Workflow Modeling, *Computer Supported Cooperative Work* **9**(3–4) (2000), 365–397.

[15] J. Eder and W. Liebhart, *The Workflow Activity Model WAMO,* 3rd International Conference on Cooperative Information Systems, 1995.

[16] C. Ellis and K. Keddara, ML-DEWS: Modeling Language to Support Dynamic Evolution within Workflow Systems, *Computer Supported Cooperative Work* **9**(3–4) (2000), 293–333.

[17] R. Fileto, L. Liu, C. Pu, E. Delgado Assad and C. Bauzer Medeiros, POESIA: An Ontological Workflow Approach for Composing Web Services in Agriculture, *VLDB Journal* **12**(4) (2003), 352–367.

[18] E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software,* Addison-Wesley, Reading, Massachusetts, 1995.

[19] D. Georgakopoulos, M. Hornick and A. Sheth, An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, *Distributed and Parallel Databases*, **3**(2) (1995), 119–153.

[20] B. Ha, J. Bae, Y.T. Park and S. Kang, Development of process execution rules for workload balancing on agents, *Data & Knowledge Engineering*, in press.

[21] M. Hammer and J. Champy, *Reengineering the Corporation*, HarperBusiness, 1993.

[22] N.R. Jennings, P. Faratin, T.J. Norman, P.O. Brien and B. Odgers, Autonomous agents for business process management, *International Journal of Applied Artificial Intelligence* **14**(2) (2000), 145–189.

[23] A. Kumar, W.M.P. van der Aalst and H.M.W. Verbeek, Dynamic Work Distribution in Workflow Management Systems: How to Balance Quality and Performance, *Journal of Management Information Systems* **18**(3) (2002), 157–193.

[24] A. Kumar and J.L. Zhao, Dynamic Routing and Operational Controls in Workflow Management Systems, *Management Science* **45**(2) (1999), 253–272.

[25] L. Liu and C. Pu, *ActivityFlow: Towards Incremental Specification and Flexible Coordination of Workflow Activities,* The 16th International Conference on Conceptual Modeling (ER '97), Los Angeles, California, USA, 1997.

[26] L. Liu and C. Pu, *Methodical Restructuring of Complex Workflow Activities,* IEEE 14th International Conference on Data Engineering, Orlando, Florida, USA, February 23–27, 1998, 342–350.

[27] L. Liu, C. Pu and D. Ruiz, Dynamic Workflow Restructuring Framework for Long-Running Business Processes, *Chapter in Advanced Topics in Database Research* **4** (2004), Idea Group Inc.

[28] K. Mark and D. Chrysanthos, A Knowledge-based Approach to Handling Exceptions in Workflow Systems, *Computer Supported Cooperative Work* **9**(3–4) (2000), 399–412.

[29] J. Mendling, G. Neumann and M. Nuttgens, *A Comparison of XML Interchange Formats for Business Process Modelling,* Proceedings of EMISA 2004 – Information Systems in E-Business and E-Government, 2004.

[30] R. Milner, A Calculus of Communicating Systems, *Lecture Notes in Computer Science* **92** (1980), Springer-Verlag, Berlin.
[31] R. Milner, *Communicating and Mobile Systems: the Pi-Calculus,* 1st edition, Cambridge University Press, June 15, 1999.
[32] R. Muller, U. Greiner and E. Rahm, AGENT WORK: a workflow system supporting rule-based workflow adaptation, *Data & Knowledge Engineering* **51**(2) (2004), 223–256.
[33] T. Murata, *Petri Nets: Properties, Analysis and Applications,* Proceedings of the IEEE, April 1989.
[34] K. Myers, *Towards a framework for continuous planning and execution,* Proceedings of the AAAI Symposium on Distributed Continual Planning, 1998.
[35] W.B. Rouse, A Theory of Enterprise Transformation, *Systems Engineering* **8**(4) (2005).
[36] Workflow Management Coalition (WfMC 1999), Terminology and Glossary 3.0 (WFMC-TC-1011). Technical report, Workflow Management Coalition, Brussels, 1999.
[37] T. Zhou, C. Pu and L. Liu, *Dynamic Restructuring of Transactional Workflow Activities: A Practical Implementation Method,* Proceedings of the 1998 International Conference on Information and Knowledge Management, Bethesda, MD, Nov. 3–7, 1998.
[38] T. Zhou, C. Pu and L. Liu, *Towards Building Flexible and Reliable Electronic Commerce Systems: A Distributed Transactional Activity Approach,* Proceedings of the First International Conference on Telecommunications and Electronic Commerce, Nashville, TN, Nov. 19–22, 1998.
[39] T. Zhou, L. Liu and C. Pu, TAM: A System for Dynamic Transactional Activity Management, *SIGMOD*, Philadelphia, Pennsylvania, USA, 1999.

James Caverlee is a Ph.D. candidate in the College of Computing at Georgia Tech and a member of the multidisciplinary Tennenbaum Institute for enterprise transformation. His research interests are generally in the areas of Web and Distributed Information Management, with an emphasis on: (1) Enterprise Computing and Workflow Management; (2) Spam-Resilient Web-Scale Computing; and (3) Web Information Retrieval and Management. James graduated magna cum laude from Duke University in 1996 with a B.A. in Economics. He received the M.S. degree in Engineering-Economic Systems & Operations Research in 2000, and the M.S. degree in Computer Science in 2001, both from Stanford University.

Joonsoo Bae is an Assistant Professor in the Department of Industrial and Information Systems Engineering at Chonbuk National University. He received the Ph.D., M.S., and B.S. degrees in Industrial Engineering from Seoul National University, South Korea in 2000, 1995, and 1993, respectively. He also completed a one-year postdoctoral course in the College of Computing at Georgia Institute of Technology in 2006. He has industry experience in LG-EDS as a technical consultant of the SCM & CRM team from 2000 to 2002. He is interested in system design and integration of management information system and e-Business technology. His research topics include business processes management using workflow systems and advanced internet application.

Qinyi Wu is a Ph.D. student in the Distributed Data Intensive Systems Lab at Georgia Institute of Technology. Her research interests include web services, workflows, and transaction processing. She received the M.S. in Computer Science from the Beijing Institute of Technology, Bejing, China. She is a member of the IEEE.

Ling Liu is an associate professor at the College of Computing at Georgia Tech. There, she directs the research programs in Distributed Data Intensive Systems Lab (DiSL), examining research issues and technical challenges in building large scale distributed computing systems that can grow without limits. Dr. Liu and the DiSL research group have been working on various aspects of distributed data intensive systems, ranging from distributed computing systems, enterprise systems to business workflow management systems. Prof. Liu has published more than 160 technical papers in the areas of Internet Computing systems, Internet data management, distributed systems, and information security. She is the recipient of best paper award of WWW 2004 and best paper award of IEEE ICDCS 2003, and a recipient of 2005 Pat Goldberg Memorial Best Paper Award. Her research group has produced a number of software systems that are either open sources or directly accessible online, among which the most popular ones are WebCQ and XWRAPElite. Dr. Liu is currently on the editorial board of several international journals, including IEEE Transactions on Knowledge and Data Engineering, International Journal of Very large Database systems (VLDBJ), International Journal of Web Services Research, and has chaired a number of conferences as a PC chair, a vice PC chair, or a general chair, including IEEE International Conference on Data Engineering (ICDE 2004, ICDE 2006, ICDE 2007), IEEE International Conference on Distributed Computing (ICDCS 2006), IEEE International Conference on Collaborative Computing (CollaborateCom 2005, 2006), IEEE International Conference on Web Services (ICWS 2004). She is a recipient of IBM Faculty Award (2003, 2006). Prof. Liu's current research is partly sponsored by grants from NSF CISE CSR, ITR, CyberTrust, AFOSR, and IBM.

Calton Pu holds the position of Professor and John P. Imlay, Jr. Chair in Software at Georgia Institute of Technology. He has published extensively in operating systems, transaction processing, and Internet data management. His current research is mainly sponsored by NSF, DARPA, Department of Energy, HP and IBM. He received his Ph.D. in Computer Science from University of Washington in 1986. He is a member of ACM, a senior member of IEEE, and a fellow of AAAS.

William B. Rouse is the Executive Director of the Tennenbaum Institute at the Georgia Institute of Technology. This university-wide center pursues a multi-disciplinary portfolio of initiatives focused on research and education to provide knowledge and skills that enable enterprise transformation. He is also a professor in the College of Computing and School of Industrial and Systems Engineering. His earlier positions include Chair of the School of Industrial and Systems Engineering, CEO of two innovative software companies – Enterprise Support Systems and Search Technology – and faculty positions at Georgia Tech, University of Illinois, Delft University of Technology, and Tufts University.

Rouse has almost four decades of experience in research, education, engineering, management, and marketing. His expertise includes individual and organizational decision making and problem solving, as well as design of organizations and information systems. In these areas, he has consulted with well over one hundred large and small enterprises in the private, public, and non-profit sectors, where he has worked with several thousand executives and senior managers.

Rouse has written hundreds of articles and book chapters, and has authored many books, including most recently *Essential Challenges of Strategic Management* (Wiley, 2001) and the award-winning *Don't Jump to Solutions* (Jossey-Bass, 1998). He is editor of *Enterprise Transformation: Understanding and Enabling Fundamental Change* (Wiley, 2006), co-editor of *Organizational Simulation: From Modeling & Simulation to Games & Entertainment* (Wiley, 2005), co-editor of the best-selling*Handbook of Systems Engineering and Management* (Wiley, 1999), and editor of the eight-volume series *Human/Technology Interaction in Complex Systems* (Elsevier). Among many advisory roles, he has served as Chair of the Committee on Human Factors of the National Research Council, a member of the US Air Force Scientific Advisory Board, and a member of the DoD Senior Advisory Group on Modeling and Simulation.

Rouse is a member of the National Academy of Engineering, as well as a fellow of four professional societies – Institute of Electrical and Electronics Engineers (IEEE), the International Council on Systems Engineering (INCOSE), the Institute for Operations Research and Management Science, and the Human Factors and Ergonomics Society. He has received the Joseph Wohl Outstanding Career Award and the Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Society; a Centennial Medal and a Third Millennium Medal from IEEE; the Best Article Award from INCOSE, and the O. Hugo Schuck Award from the American Automation Control Council. He is listed in *Who's Who in America, Who's Who in Engineering,* and other biographical literature, and has been featured in publications such as *Manager's Edge*, *Vision*, *Book-Talk*, *The Futurist*, *Competitive Edge*, *Design News*,*Quality & Excellence, IIE Solutions*, *Industrial Engineer*, and *Engineering Enterprise.*

Rouse received his B.S. from the University of Rhode Island, and his S.M. and Ph.D. from the Massachusetts Institute of Technology.