Distributed and Parallel Databases, 3, 119–153 (1995)

An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure

DIIMITRIOS GEORGAKOPOULOS AND MARK HORNICK {dimitris,mfh0}@gte.com
GTE Laboratories Incorporated, 40 Sylvan Road, Waltham, MA 02254

AMIT SHETH amit@cs.uga.edu
University of Georgia, LSDIS Lab, Department of C.S., 415 GSRC, Athens, GA 30602

Recommended by: Omran Bukhres and e. Kühn

Abstract. Today's business enterprises must deal with global competition, reduce the cost of doing business, and rapidly develop new services and products. To address these requirements enterprises must constantly reconsider and optimize the way they do business and change their information systems and applications to support evolving business processes. Workflow technology facilitates these by providing methodologies and software to support (i) business process modeling to capture business processes as workflow specifications, (ii) business process reengineering to optimize specified processes, and (iii) workflow automation to generate workflow implementations from workflow specifications. This paper provides a high-level overview of the current workflow management methodologies and software products. In addition, we discuss the infrastructure technologies that can address the limitations of current commercial workflow technology and extend the scope and mission of workflow management systems to support increased workflow automation in complex real-world environments involving heterogeneous, autonomous, and distributed information systems. In particular, we discuss how distributed object management and customized transaction management can support further advances in the commercial state of the art in this area.

Keywords: Business Process Re-engineering, Workflow Systems, Customized Transaction Management

1. Introduction

The workflow concept has evolved from the notion of process in manufacturing and the office. Such processes have existed since industrialization and are products of a search to increase efficiency by concentrating on the routine aspects of work activities. They typically separate work activities into well-defined tasks, roles, rules, and procedures which regulate most of the work in manufacturing and the office. Initially, processes were carried out entirely by humans who manipulated physical objects. With the introduction of information technology, processes in the work place are partially or totally automated by information systems, i.e., computer programs performing tasks and enforcing rules which were previously implemented by humans.

Medina-Mora et al. [32] categorize processes in an organization into material processes, information processes, and business processes. The scope of a *material process* is to assemble physical components and deliver physical products. That is, material processes relate human tasks that are rooted in the physical world. Such tasks include, moving, storing, transforming, measuring, and assembling physical objects.

Information processes relate to automated tasks (i.e., tasks performed by programs) and partially automated tasks (i.e., tasks performed by humans interacting with computers) that create, process, manage, and provide information. Typically an information process is rooted in an organization's structure and/or the existing environment of information systems. Database, transaction processing, and distributed systems technologies provide the basic infrastructure for supporting information processes.

Business processes are market-centered descriptions of an organization's activities, implemented as information processes and/or material processes. That is, a business process is engineered to fulfill a business contract or satisfy a specific customer need. Thus, the notion of a business process is conceptually at a higher level than the notion of information or material process. In this paper, we focus on business processes that are primarily implemented as information processes.

Once an organization captures its business in terms of business processes, it can reengineer each process to improve it or adapt it to changing requirements. Reasons cited for business process redesign include increasing customer satisfaction, improving efficiency of business operations, increasing quality of products, reducing cost, and meeting new business challenges and opportunities by changing existing services or introducing new ones. *Business process reengineering* involves explicit reconsideration and redesign of the business process. It is performed before information systems and computers are used for automating these processes. *Information process reengineering* is a complementary activity of business process reengineering. It involves determining how to use legacy and new information systems and computers to automate the reengineered business processes. The two activities can be performed iteratively to provide mutual feedback. While business process redesign can explicitly address the issues of customer satisfaction, the information process reengineering can address the issues of information system efficiency and cost, and take advantage of advancements in technology.

Workflow is a concept closely related to reengineering and automating business and information processes in an organization. A workflow may describe business process tasks at a conceptual level necessary for understanding, evaluating, and redesigning the business process. On the other hand, workflows may capture information process tasks at a level that describes the process requirements for information system functionality and human skills. The distinction between these workflow perspectives is not always made, and sometimes the term *workflow* is used to describe either, or both, of the business and information systems perspectives.

Workflow management (WFM) is a technology supporting the reengineering of business and information processes. It involves:

1. defining *workflows*, i.e., describing those aspects of a process that are relevant to controlling and coordinating the execution of its tasks (and possibly the skills of individuals or information systems required to perform each task), and
2. providing for fast (re)design and (re)implementation of the processes as business needs and information systems change

To effectively support WFM, organizations must evolve their existing computing environments to a new distributed environment that:

- is *component-oriented*, i.e., supports integration and interoperability among loosely-coupled components corresponding to heterogeneous, autonomous, and/or distributed (HAD) legacy and new systems,
- supports *workflow applications* corresponding to business or information process implementations accessing multiple HAD systems,

- ensures the correctness and reliability of applications in the presence of concurrency and failures, and
- supports the evolution, replacement, and addition of workflow applications and component systems as processes are reengineered.

Many commercial systems have been introduced to support WFM. The genesis of WFM software was probably in automating document-driven business processes [40]. Some of the early products were extensions to the document imaging and management software [4]. Rosy estimates of fast expanding market size from less than $100 million in 1991 to about $2.5 billion in 1996 [27] drew significant interest of software companies, and spawned a host of new products for WFM. Presently, commercial WFM systems for office automation can support document management, imaging, application launching, and/or human coordination, collaboration, and co-decision. Although many of these WFM systems meet some of the requirements above, they allow only limited interoperability (in terms of the types of HAD systems they can integrate and tasks they support), may not ensure correctness or reliability of applications in the presence of concurrency and failures, and suffer from performance and scalability problems. Therefore, commercial WFM systems currently cannot support enterprise-wide workflow applications effectively.

To satisfy these requirements, we believe that the following two key infrastructure technologies must be combined with the capabilities commercial WFM systems already provide:

- distributed object management (computing)
- customized transaction management

Distributed object management (DOM) [30, 35, 36] supports the interoperability and integration of HAD systems and applications implementing business or information processes. DOM allows WFM systems to cope with replacement, migration, and evolution of HAD systems or changes in their functionality and data. In addition, DOM provides an object model that facilitates managing complexity by the use of abstraction, inheritance, and polymorphism. Other distributed computing approaches that currently offer a lower level of interoperability than DOM may also be useful in providing interoperability for WFM.

Customized transaction management (CTM) ensures the correctness and reliability of applications implementing business or information processes, while permitting the functionality each particular process requires (e.g., isolation, coordination, or collaboration between tasks). In addition, CTM copes with changes in (i) the correctness and reliability requirements of the process, and (ii) the correctness and reliability guarantees HAD systems provide.

In this paper, we discuss WFM technology from process specification to workflow implementation. While the emphasis is more on the state-of-the-art in commercial WFM systems, we also discuss the infrastructure technologies we believe can address the fundamental limitations found in today's commercial WFM systems. In particular, we describe only the technologies we believe can significantly improve the WFM technology and do not attempt to comprehensively review all related research. Furthermore, we strive to give a balanced treatment to both process and data management issues. The paper by Krishnakumar and Sheth [24] provides a more detailed discussion of a workflow model, an "intermediate" specification language, and a system architecture to support workflow automation in an environment consisting of HAD systems.

The structure of this paper is as follows: In Section 2, we define WFM in greater detail. In Section 3, we describe the principles of process modeling, and give examples. In

Section 4, we discuss workflow specification and implementation as currently supported by commercial WFM systems, as well as the limitations of WFM products. In Section 5, we discuss key infrastructure technologies for WFM and describe research issues and corresponding work in progress. Specifically, we discuss the importance of DOM and CTM technologies for the advancement of WFM technology. Our conclusions and some perspective on future expectations are presented in Section 6.

2. Workflows and workflow management

There is little agreement as to what workflow is and which features a workflow management system must provide. Under the umbrella of the term "workflow", which is often used casually, people may be referring to a business process, specification of a process, software that implements and automates a process, or software that simply supports the coordination and collaboration of people that implement a process. Various concepts attributed to the term workflow are illustrated in Fig. 1.

For example, consider the following definitions of workflow from software vendors which produce workflow products:

- A representative of PeopleSoft Inc. states that "Workflow is the mechanism by which you can implement business reengineering practices" [14].
- Product literature from Action Technologies Inc. defines workflow as "work [that] is recast as a series of people-based transactions" and states that "A series of workflows form a business process" [14].
- Product literature from Recognition Internal Inc. states that "simply defined, [workflow] is the process by which individual tasks come together to complete a transaction—a clearly defined business process—within an enterprise" [3].
- A Wang Laboratories representative states that "Workflow goes beyond routing [i.e., moving information among users or systems] by integrating information from a variety of sources" [14].

Other definitions of workflow distinguish workflow specification from workflow implementation. For example, in [39] workflows are defined as activities involving the coordinated execution of multiple tasks performed by different processing entities. A task defines some work to be done and can be specified in a number of ways, including a textual description in a file or an electronic mail message, a form, or a computer program. A

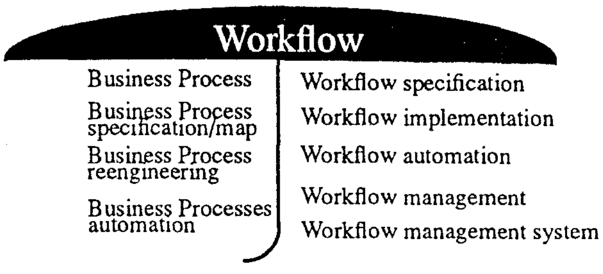
Workflow
Business Process
Business Process specification/map
Business Process reengineering
Business Processes automation
Workflow specification
Workflow implementation
Workflow automation
Workflow management
Workflow management system

Figure 1. The "workflow umbrella."

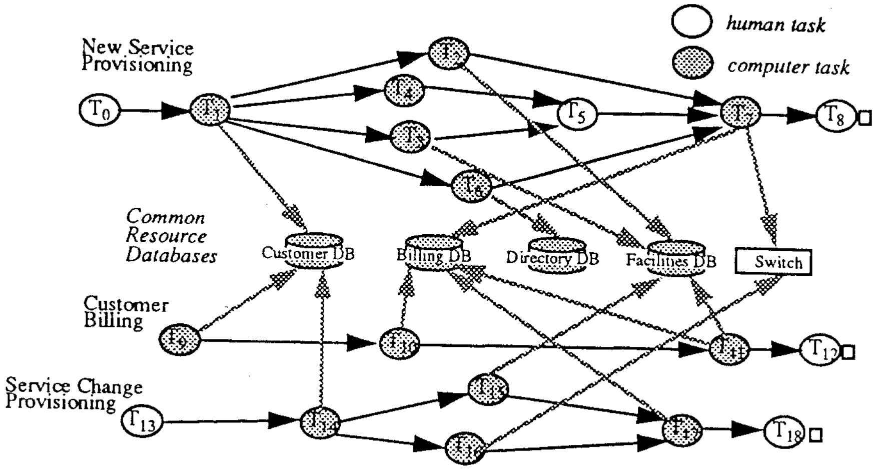

Figure 2. Telecommunication workflows.

processing entity that performs the tasks may be a person or a software system (e.g., a mailer, an application program, a database management system). Specification of a workflow involves describing those aspects of its constituent tasks (and the processing entities that execute them) that are relevant to controlling and coordinating their execution. It also requires specification of the relationships (i.e., dependencies) among tasks and their execution requirements. These can be specified using a variety of software paradigms (e.g., rules, constraints, or programs).

The above workflow definitions do not clarify the relationship of the terms under the workflow umbrella in Fig. 1. In the following sections we provide definitions of these concepts and give examples.

2.1. *Workflows and workflow systems*

We define a *workflow* as a collection of *tasks* organized to accomplish some business process (e.g., processing purchase orders over the phone, provisioning telephone service, processing insurance claims). A task can be performed by one or more software systems, one or a team of humans, or a combination of these. Human tasks include interacting with computers closely (e.g., providing input commands) or loosely (e.g., using computers only to indicate task progress). Examples of tasks include updating a file or database, generating or mailing a bill, and laying a cable. In addition to a collection of tasks, a workflow defines the order of task invocation or condition(s) under which tasks must be invoked, task synchronization, and information flow (dataflow).

Figure 2 depicts three telecommunication workflows which require accesses to some combination of shared databases. The New Service Provisioning workflow captures the process of telephone service provisioning for a new customer. The workflow takes place when a telephone company customer requests telephone service installation.

Task T_0 involves an operator collecting information from the customer. When sufficient customer data are collected, task T_1 is performed to (i) verify whether the information provided by the customer is accurate and (ii) create a corresponding service order record. On completion of T_1, tasks T_2, T_3, and T_4 are initiated to perform three line provisioning activities. The objective of a provisioning activity is construct a circuit from a customer location to the appropriate telephone switch and allocate equipment to connect the circuit. Only one of these provisioning tasks should be allowed to complete, as all will result in a completed circuit, i.e., a set of lines and equipment that connects the customer to a telephone network (this requirement is not depicted in Fig. 2). T_2 attempts to provide a connection by using existing facilities such as lines and slots in switches. If T_2 succeeds, the cost of provisioning is minimal, i.e., the requested connection can be established by allocating existing resources. However, a successful completion of this task may not be possible if the facilities are not available. T_3 and T_4 achieve the same objectives as T_2 but involve different paths for physical installations of new facilities. T_5 requires manual work for facility installation. The human task T_5 is initiated by providing installation instructions to the engineers (e.g., via hand-held terminals) and is completed when the human engineers provide the necessary work completion data. Task T_6 involves changes in the telephone directory, while T_7 updates the telephone switch to activate service and then generates a bill. Finally, task T_8 involves a human operator who calls the customer to inform him of the establishment of the requested service and verify that the provided service meets the customer needs. In addition to the tasks involved, the workflow defines the following task dependencies: (i) T_1 starts after the completion of T_0, (ii) T_2, T_3, T_4, and T_6, can be performed concurrently after task T_1 is completed, (iii) T_5 must start after the completion of T_3 and T_4, (iv) T_7 is performed after the completion of T_2, T_5, and T_6, and (v) T_8 starts after T_7 completes.

The other workflows depicted in Fig. 2 represent other telephone operations activities. Their tasks and the task sequencing have explanations similar to that of the New Service Provisioning workflow.

A number of organizations have produced *workflow management systems* (WFMSs) that provide the ability to specify, execute, report on, and dynamically control workflows involving multiple humans and HAD systems [31]. The capabilities commercial WFMSs currently offer are discussed in Section 4 and Appendix A.

2.2. *Characterizing workflows*

As yet, there is no commonly agreed way to characterize or categorize workflows or WFMSs. Furthermore, most workflow characterizations neglect highly automated workflows accessing a large number of shared information systems. In Section 2.2.1, we describe workflows and WFMSs as they are characterized by the trade press. In Section 2.2.2, we discuss another characterization of workflows that considers highly automated workflows accessing shared information systems and emphasizes workflow implementation and automation requirements.

2.2.1. Trade press workflow characterization. The trade press often distinguishes between three kinds of workflow (this characterization was first given by McCready [27]): *ad hoc*, *administrative*, and *production*. The dimensions along which these kinds of workflow are often described include:

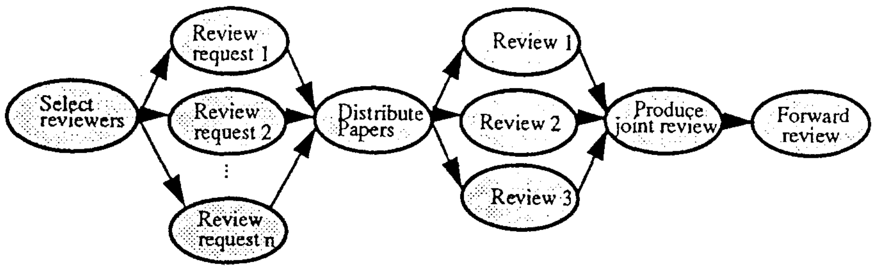

Figure 3. Ad hoc paper review workflow.

- repetitiveness and predictability of workflows and tasks
- how the workflow is initiated and controlled, e.g., from human-controlled to automated
- requirements for WFMS functionality

Ad hoc workflows perform office processes, such as product documentation or sales proposals, where there is no set pattern for moving information among people [23, 4]. Ad hoc workflow tasks typically involve human coordination, collaboration, or co-decision [41]. Thus, the ordering and coordination of tasks in an ad hoc workflow are not automated but are instead controlled by humans. Furthermore, the task ordering and coordination decisions are made while the workflow is performed. Ad hoc workflows typically involve small teams of professionals and are intended to support short term activities which require a rapid workflow solution, e.g., supporting the process of putting together the program of a professional conference.

WFMSs that support ad hoc workflows must provide functionality for facilitating human coordination, collaboration, and co-decision. Functionality for controlling task ordering is typically not provided in such WFMSs. Users of an ad hoc workflow need to access the WFMS to determine if work was completed. Also, ad hoc WFMSs are not mission critical, i.e., periodic failure of such workflows does not significantly interfere with the overall business process. The infrastructure technology currently used by ad hoc WFMSs ranges from "enhanced" electronic mail to group calendaring and conferencing systems. Ad hoc WFMSs usually use a (proprietary) database to store shared information (e.g., documents such as conference review forms or papers). WFMSs that support ad hoc workflow are also called *groupware.*

Figure 3 illustrates a simplified ad hoc workflow involving the review process for conference papers. The review process is to select reviewers, distribute the paper(s) to the selected reviewers, have the reviewers perform the reviews and collaborate in producing a joint review document, and finally forward it to the authors. This is an ad hoc workflow because it involves: (i) negotiation for selecting the reviewers, and (ii) collaboration between the reviewers for producing a joint review. Furthermore, subsequent paper reviews may not be performed by the same reviewers. Thus, ad hoc workflows usually set up procedures for performing one-of-a-kind activities.

Administrative workflows involve repetitive, predictable processes with simple task coordination rules, such as routing an expense report or travel request through an authorization process. The ordering and coordination of tasks in administrative workflows can be automated. WFMS that support administrative workflow handle simple information routing and document approval functions, such as those found in travel planning and purchase

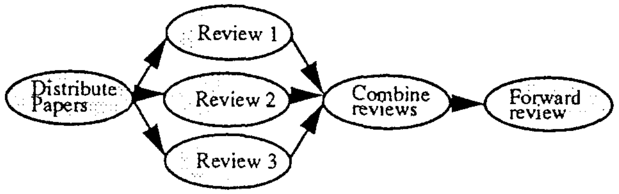

Figure 4. Administrative paper review workflow.

requests. Administrative workflows do not encompass a complex information process and do not require accesses to multiple information system used for supporting production and/or customer services. Administrative WFMS are generally non-mission critical. The infrastructure technology they currently use is typically based on electronic mail.

Consider again the paper review process. However, this time assume that the reviewers are known in advance (e.g., the same reviewers are used for all paper reviews). Furthermore, suppose that the reviewers do not collaborate in producing a joint review. Instead, they produce individual reviews that are considered by the editor (e.g., program chair) who makes the final decision. Under these assumptions the paper review workflow becomes an administrative workflow such as depicted in Fig. 4.

In an administrative workflow users are actively *prompted* to perform their tasks. Whereas reviewers using an ad hoc workflow needed to access the WFMS to determine if work was completed, reviewers using an administrative WFMS may receive email with review instructions along with the paper to be reviewed and a reviewer's comments form. When the form is completed, it is automatically routed to the program committee chairperson who is alerted when all of the reviews are completed.

Production workflows involve repetitive and predictable business processes, such as loan applications or insurance claims. Unlike administrative workflow, production workflows typically encompass a complex information process involving access to multiple information systems. The ordering and coordination of tasks in such workflows can be automated. However, automation of production workflows is complicated due to: (i) information process complexity, and (ii) accesses to multiple information systems to perform work and retrieve data for making decisions (administrative workflows rely on humans for most of the decisions and work performed).

WFMSs that support production workflow must provide facilities to define task dependencies, and control task execution with little or no human intervention. Production WFMSs are often mission critical and must deal with the integration and interoperability of HAD information systems.

Consider the simplified health claims process workflow depicted in Fig. 5. In the health claims workflow, a claim form is first manually scanned and stored into an object database. Then the claim is manually indexed in a relational database. This information is subsequently analyzed by an automated "Assess Claim" task. The task is performed by an expert system which uses an eligibility database to determine if payment should be made. If the claim is rejected, a claim representative discusses the claim with the customer and either agrees to make some payment, or to reject the claim. If payment is made, the "make payment" task accesses the finance database and records the payment. The significant differences between this production workflow and either the ad hoc or administrative workflow

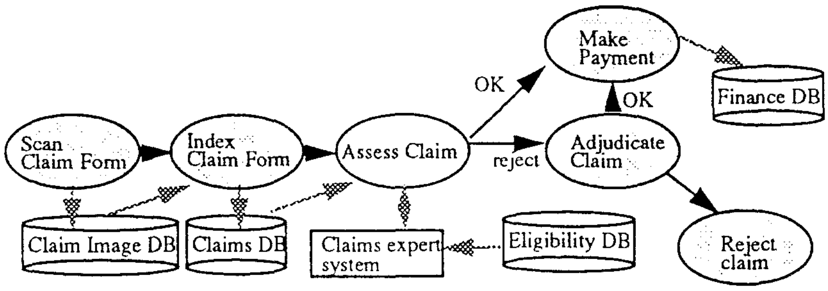

Figure 5. "Health claims process" workflow.

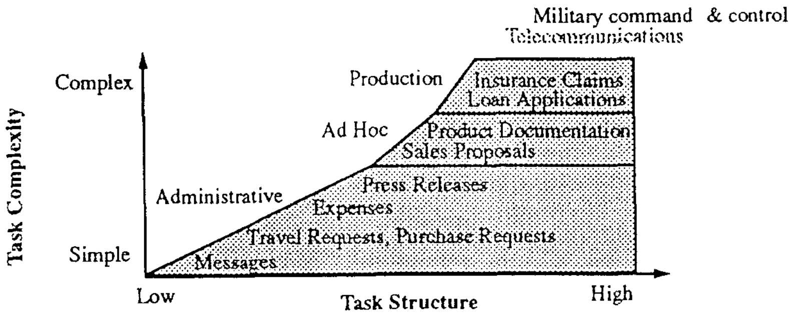

Figure 6. Trade press characterization of workflow.

are: (i) the interaction of information systems with the business process, and (ii) the use of automated (non-human) task performers.

The relationship of ad hoc, administrative, and production workflow is illustrated in Fig. 6 [23] using *task structure* versus *complexity.*

Workflow with little structure may involve a linear path of tasks to be followed; highly structured workflow may involve a graph-like organization of tasks where some tasks may be executed in parallel or multiple tasks must complete before others can start. Complexity can be determined by the kinds of coordination/collaboration rules or constraints applied to task execution. For example, one aspect of complexity could be a requirement that a task begins execution only after a set of events has occurred. Complexity is also reflected by the kinds of HAD systems that must be integrated to produce a task implementation, e.g., office applications, DBMSs, or legacy information systems.

Other characterizations of workflows have recently appeared in the trade press [1, 14]. [1] divides workflows into ad hoc workgroup support, task automation, document flow, and process automation. [14] divides workflows into three categories: mail-centric, document-centric and process-centric. These characterizations do not separate workflow semantics from the commercial WFMS that support them, and the infrastructure technology they are currently using. Furthermore, trade press workflow characterizations typically do not distinguish between production workflows accessing a small number of homogeneous information systems and highly automated workflows accessing many shared HAD information systems. For example, workflows in the domains of military command and control, or telecommunications belong in the latter category. Such workflows are not discussed or characterized by the trade press. Using the complexity versus structure framework in

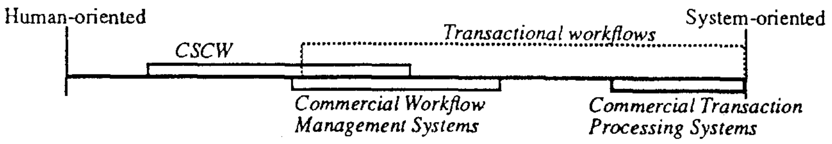
Human-oriented
System-oriented
CSCW
Transactional workflows
Commercial Workflow Management Systems
Commercial Transaction Processing Systems

Figure 7. Characterizing workflow.

Fig. 6, these workflows have requirements with greater structure and complexity than those found in production workflows. Since workflows in domains such as military command and control or telecommunications involve HAD systems with greater heterogeneity (e.g., controlling telecommunications switch hardware in telecommunications) and greater demands for correct and reliable execution (e.g., military applications on whose data integrity lives depend), their implementation and automation requirements are greater than those in production workflows. As a result we now characterize workflows according to their implementation and automation requirements.

2.2.2. *Our characterization of workflow.*

We characterize workflow along a continuum from human-oriented to system-oriented as depicted in Fig. 7. On the one extreme, *human-oriented* workflow involves humans collaborating in performing tasks and coordinating tasks. The requirements for WFMSs in this environment are to support the coordination and collaboration of humans and to improve human throughput. Humans, however, must ensure the consistency of documents and workflow results.

On the other extreme, *system-oriented* workflow involves computer systems that perform computation-intensive operations and specialized software tasks. In addition to being highly automated, system-oriented workflows access HAD information systems. While human-oriented workflow implementations often control and coordinate[1] human tasks, system-oriented workflow implementations control and coordinate software tasks (typically with little or no human intervention). Consequently, system-oriented workflow implementations must include software for various concurrency control and recovery techniques to ensure consistency and reliability. This is not required and cannot be provided by WFMS that support human-oriented workflows. Human-oriented workflows have process semantics (e.g., capture where to route a document) but have no real knowledge of the (semantics of) the information being processed. Therefore, in human-oriented workflows the WFMS is there to assist people and the WFMS cannot be made responsible for maintaining data consistency, since it has no information semantics. On the other hand, system-oriented workflows have more knowledge of information semantics (e.g., built-into the various applications involved and the information systems that synchronize application access to shared databases). Hence the WFMS can be given (and must be given) more responsibility for maintaining information consistency.

In human-oriented workflow, the main issues to address include:

- human-computer interaction
- matching human skills to task requirements
- changing office culture, i.e., how people need or prefer to work

In systems-oriented workflow, the issues to address include:

- matching business process requirements to functionality and data provided by existing information systems and/or their applications

- interoperability among HAD systems
- finding appropriate software tasks to perform workflow tasks
- determining new software required to automate business processes
- ensuring correct and reliable system execution

Issues such as exception handling, user overrides, prioritization, and deadline may appear in different forms in both types of system, and need to be addressed. Also depicted in Fig. 7 are segments indicating a range of workflow characteristics and issues that are addressed by (i) the field of computer supported cooperative work (CSCW), (ii) commercial WFMSs, and (iii) commercial transaction processing (TP) systems (e.g., distributed DBMSs, TP monitors). CSCW overlaps with WFM where workflows involve predominantly human tasks. Commercial TP systems overlap with WFM when workflow applications are submitted as DBMS or TP monitor transactions.

Transactional workflows involve coordinated execution of multiple tasks that (i) may involve humans, (ii) require access to HAD systems, and (iii) support selective use of transactional properties (e.g., atomicity, consistency, isolation, and/or durability) for individual tasks or entire workflows. Selective use of transactional properties is required to allow the specialized funtionality required by each workflow (e.g., allow task collaboration and support complex workflow structures). Since the traditional transactions DBMS and TP monitor provide do not permit selective use of transactional properties to allow specialized functionality (e.g., DBMS-provided transactions enforce isolation and this does not permit the cooperation of tasks or workflows,) extending or relaxing the transaction models DBMS and TP monitors provide is needed to support workflow functionality requirements. WFMSs currently do not address key aspects of system-oriented workflow such as HAD systems and WFMS interoperability and integration, and ensuring correct and reliable workflow execution in the presence of concurrency and failures. Transactional workflows and the technology needed to support these are discussed further in Section 5.2.

2.3. *Workflow management*

In the rest of this paper we use the generic term *process* to refer to a business process, a business process and its corresponding information process, or an information process. Workflow management involves everything from modeling processes up to synchronizing the activities of information systems and humans that perform the processes. In particular, management of a workflow includes the following (as illustrated in Fig. 8):

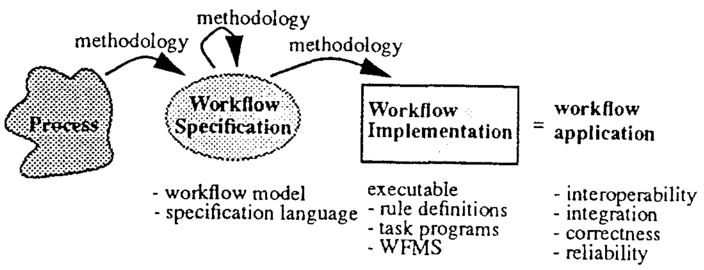

Figure 8. Workflow management issues.

1. *process modeling and workflow specification*: requires workflow models and methodologies for capturing a process as a workflow specification,
2. *process reengineering*: requires methodologies for optimizing the process, and
3. *workflow implementation and automation*: requires methodologies/technology for using information systems, and human performers to implement,schedule, execute, and control the workflow tasks as described by the workflow specification.

The following paragraphs discuss each of these workflow management issues.

Modeling a process. Before we capture a process we first need to understand it. This usually involves interviewing people with expert knowledge about the process. Interview methodologies such as those used for expert system design are appropriate for conducting such interviews. When enough knowledge about the process is obtained, workflow specification is performed to capture the process.

GTE Telephone Operations is performing a large process reengineering effort [11]. GTE Telephone Operations formed reengineering teams (20–25 employees) to capture existing business processes and redesign its core business processes. Teams documented existing business processes by conducting 1000 interviews and 10,000 observations, and produced corresponding workflow specifications using a workflow specification tool.

In addition to understanding a business process, modeling the process involves workflow specification. A workflow specification captures a process abstraction. The process abstraction level in a workflow specification depends on the intended use of the workflow specification. For example, a workflow specification may describe a process at the highest conceptual level necessary for understanding, evaluating, and redesigning the process. On the other hand, another workflow specification may describe the same process at a lower-level of detail required for performing workflow implementation.

Performing workflow specification requires a *workflow model.* A workflow model typically includes a set of concepts that are useful to describe processes, their tasks, the dependencies among tasks, and the required *roles* (i.e., skills of the individuals or information systems) that can perform the specified tasks. Workflow models are discussed further in Section 3.

Workflow specification is typically performed using a *workflow specification language.* Workflow specification languages in commercial WFMS use rules, constraints, and/or graphical constructs to describe the ordering and synchronization of tasks in a workflow, and task attributes to describe the tasks and the roles to perform them. For example, a graphical workflow specification may be similar to the illustration in Fig. 2 (possibly without including the common resource databases unless they indicate the roles of the information systems and the humans required to implement the specified tasks). An example rule in a rule-based specification of the New Service Provisioning process in Fig. 2 might be: *On* T_1 *completion Do start* T_2, T_3, T_4, T_6. This rule captures the fact that in tasks T_2, T_3, T_4, and T_6 must be executed after the completion of task T_1.

Reengineering a process. The objective of re-engineering methodologies is to optimize business processes. Process optimization strategies depend on the reengineering objectives (e.g., increasing customer satisfaction, reducing the cost of doing business, introducing new products or services). Reengineering methodologies are currently an art. Workflow specification provides a high-level description of a process that facilitates high-level reasoning about business process efficiency.

An effort undertaken at NYNEX to reengineer the process for provisioning of requests for T-1 lines is reported in [34]. The GTE Telephone Operations reengineering effort and the RAPID methodology used for improving customer service and reduce the information system costs are described in [11].

Implementing and automating a workflow. Implementation deals with the issues associated with realizing a workflow using computers, software, information systems, and/or WFMSs. Workflow automation deals with scheduling and controlling workflow execution.

No workflow implementation or automation is required when the *only* reason for workflow specification is to capture business processes and reason about their efficiency. Otherwise, workflow specifications are used to implement and automate workflows. In particular, workflow specification and implementation and can be loosely coupled (e.g., workflow specifications are implemented by software engineers) or tightly coupled (e.g., workflow specifications are provided as direct input to a WFMS that either generates code or interprets specifications for controlling workflow execution).

Many commercial workflow management systems take the tightly coupled approach from workflow specification to workflow implementation. The implication is that the dividing line between what is the workflow specification and workflow implementation is not always sharp.

Automated workflow implementations are typically distributed applications that access HAD systems. Like any other distributed HAD system application, a WFMS has to deal with application integration, interoperability, and implementation correctness and reliability. Limitations of WFMSs in dealing with these issues, and infrastructure technologies that can be used to address these limitations, are discussed in Section 5.

3. Process modeling and workflow specification

Process modeling involves capturing a process in a workflow specification. In this section, we discuss workflow models and corresponding process modeling methodologies.

3.1. Methodologies for process modeling

There are two basic categories of process modeling methodologies: *communication-based* and *activity-based* [29].

The *communication-based methodologies* stem from Winograd/Flores "Conversation for Action Model" [42]. This methodology assumes that the objective of business process reengineering is to improve customer satisfaction. It reduces every action in a workflow to four phases based on communication between a *customer* and a *performer* (illustrated in Fig. 9):

1. preparation—a customer requests an action to be performed or a performer offers to do some action
2. negotiation—both customer and performer agree on the action to be performed and define the terms of satisfaction

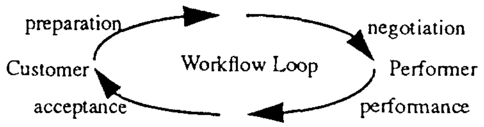

Figure 9. Conversation for Action Model.

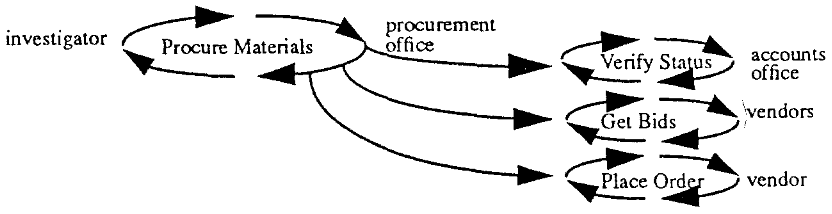

Figure 10. Workflow for procuring materials.

3. performance—the action is performed according to the terms established
4. acceptance—the customer reports satisfaction (or dissatisfaction) with the action

Each *workflow loop* between a customer and performer can be joined with other workflow loops to complete a business process. The performer in one workflow loop can be a customer in another workflow loop. The resulting business process reveals the social network in which a group of people, filling various roles, fulfill a business process.

The example in Fig. 10 illustrates a business process for procuring materials. The main workflow loop (procure materials) requires several secondary workflow loops during the performance phase (verify status, get bids, place order). In particular, an investigator requests services from the procurement office for materials. In the performance of procurement, the procurement office instructs the accounts office to verify the account status of the purchaser. The procurement office then contacts vendors for bids, and finally selects a vendor to place an order. The workflow is completed (i.e., the main loop is closed) when the procurement office reports to the investigator that the materials have been procured. Note that the performer in the main loop is the customer in the secondary loops. Also note that workflow specifications using this methodology do not indicate which activities can occur in parallel or if there are conditional or alternative actions. Since this methodology assumes that the objective of business process re-engineering is to improve customer satisfaction, the emphasis is on the customer. However, there are business processes where the customer emphasis may be superficial, e.g., if the objectives are to minimize information system cost or reduce waste of material in a process. Therefore, this methodology is not appropriate for modeling business processes with objectives other than customer satisfaction. Another limitation is that this methodology by itself does not support the development of workflow implementations for specifications.

The "ActionWorkflow Analyst" tool [33, 3] from Action Technologies is based on the Winograd/Flores model, as is the "Business Transformation Management" tool from Business Transaction Design [29].

Activity-based methodologies focus on modeling the work instead of modelling the commitments among humans. For example, consider the workflow depicted in Fig. 11 where the process "procure materials" is composed of several tasks. The arrows indicate the sequential nature of this process map. Note that "procure materials" may be a task in yet

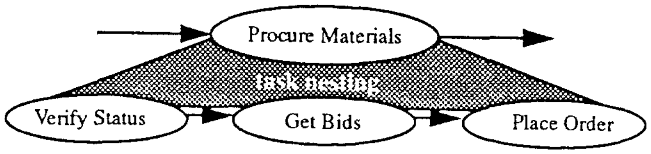

Figure 11. Workflow for procuring materials.

another workflow, and that tasks may nest arbitrarily deeply. Unlike communication-based methodologies, activity-based methodologies do not capture process objectives such as customer satisfaction.

Many commercial WFMS provide activity-based workflow models. For example, in the workflow model supported by InConcert [31], workflows (referred to as jobs) consist of tasks. Each task may be comprised from subtasks. Each task has dependencies on other tasks at the same level and has an assigned role which is the proxy for a human or a program that performs the task. GTE's RAPID methodology [11] is also activity-based. RAPID provides two workflow models: a high-level model for performing conceptual business process analysis and a lower-level model for describing the corresponding information process. In the high-level workflow model, workflows (referred as process maps) contain tasks (referred as steps) necessary to perform a particular business process. These steps can be partially or totally ordered as necessary to indicate alternatives or parallel execution of business process steps.

The communication-based and activity-based workflow models can be combined when process re-engineering objectives are compatible with both models (e.g., satisfy the customer by minimizing workflow tasks and human roles). For example, the workflow model used for the telecommunications workflows in Fig. 2 can be viewed as both activity-based and communication-based.

Object-oriented methodologies, such as those proposed in Rumbaugh et al. [38] and Jacobson [22] may be useful in defining workflow specifications (and deriving implementations). For example, Jacobson [22] describes how to (i) identify objects that correspond to "actors" (i.e., workflow roles), (ii) identify the dependencies between those objects, (iii) use object techniques such as inheritance to organize object specifications, and (iv) describe "use cases" which are essentially a sequence of tasks needed to complete some business process. Use cases may also include "alternative courses" which describe how to handle exceptional conditions. However, object orientation provides no explicit support (e.g., workflow model) for process modeling. The object designer typically must define workflow model-specific objects from scratch. This problem can be addressed if workflow-model-specific types and classes (e.g., customer, employee, document, computer system, workflow, step, etc.) are defined to support business process modeling directly.

Some commercial business process modeling tools use object oriented concepts and techniques for representing, as well as implementing processes. For example, InConcert [31] and ObjectFlow [21] combine object-orientation with the activity-based methodology.

3.2. Technology status and research issues

Many commercial WFMSs, including those discussed in this section, support workflow definition for process modeling. However, to facilitate the use of workflow across vendor products, a standard representation for workflow specification is necessary. A standards body called the *Work Flow Management Coalition* was formed in 1993 to address the lack of standards for WFMSs. Their objectives include standardizing workflow model specifications to allow the interoperability of workflow specifications supported by different WFMSs. Interoperability of workflow specifications is an open research issue.

Another problem with current commercial technology is that the workflow models and process modeling methodologies do not explicitly support the specification of what it means for a workflow to be correct, e.g., what tasks must complete for the workflow to be considered successful. For example, in a telecommunications service provisioning domain, a workflow may be considered successful if a bill is produced, even if the directory is not updated to indicate customer changes. Research in the area of transaction management can provide modeling constructs for augmenting existing workflow models to address correctness and reliability.

Finally, process modeling methodologies have not addressed workflow implementation involving legacy information systems. For organizations that rely on legacy information systems, workflow specification for performing workflow implementation requires mapping workflow specifications to legacy system functionality and data. If this is not done, process reengineering may produce workflow specifications that cannot be supported by the legacy information systems.

4. Workflow implementation and automation

As we briefly discussed in Section 2.3, many commercial WFMSs have taken the tightly coupled approach from workflow specification to workflow implementation, i.e., they are using workflow specifications to produce corresponding workflow implementations. This approach is a powerful paradigm for process implementation, since it eliminates the dividing line between workflow specification and implementation. In this section we focus on the tightly coupled approach and discuss the capabilities and limitations of commercial WFMSs.

In particular, in Section 4.1. we describe the capabilities currently supported by commercial WFMSs (a partial list of WFMS vendors and products can be found in Appendix A). In Section 4.2, we discuss commercial WFMS limitations, particularly in the areas of integration with HAD information systems and support for workflow correctness and reliability. In Section 5, we discuss infrastructure technologies that can complement the capabilities of commercial WFMS to address these limitations.

4.1. Commercial workflow management systems

In this section we discuss the features and capabilities currently supported by commercial WFMSs with respect to workflow model, specification language, tools for testing/analysis and monitoring, system architectures and interoperability, implementation support, and correctness and reliability.

Workflow model. WFMSs provide both activity-based and communication-based workflow models for specifying workflows. For example, InConcert, Staffware, and FloWare use activity-based workflow models, while ActionWorkflow uses a communication-based model.

The workflow models most WFMSs support are activity-based, and consist of elements similar to the following:

- *workflows*: a partial or total ordering of a set of tasks
- *tasks*: a partial or total order of operations, descriptions for human actions, or other tasks
- *manipulated objects*: documents, data records, images, phones, fax machines, printers, etc.
- *roles*: a placeholder for a human skill or an information system service required to perform a particular task
- *agents*: humans or information systems that fill roles, perform tasks, and interact during workflow execution

To provide different levels of abstraction, WFMSs typically support the nesting of tasks. For example, the workflow that provides a new customer with telephone service involves tasks of acquiring customer information, allocating facilities, and setting up customer billing. The task for allocating facilities may in turn be comprised of several line provisioning sub-tasks that explore different line provisioning alternatives (e.g, use of existing facilities or installation of new facilities). Each level of abstraction provides a view to the workflow specification. Higher levels of abstractions help management follow or control a business process. The lower levels of abstraction are required to capture exactly what is required to implement a workflow.

The definition of roles in a workflow is particularly beneficial when a task can be performed by more than one agent. The mapping of agents to roles (role/user administration in Appendix B) helps to manage change in the work force and the computing environment. In addition, it can facilitate dynamic load balancing. For example, if the role is 'PurchaseOrderApprover', a purchasing department may have several users (human agents) who can fill this role. If the workload on one PurchaseOrderApprover is high, the system can automatically pass a request to another PurchaseOrderApprover.

Specification language. All WFMSs of which we are aware provide graphical workflow specification languages. In addition, many WFMSs provide rule-based or constrained workflow specification languages. These languages are higher-level languages than standard programming languages such as C and C++. They support the specification of the following:

- task structure (control flow) and information exchange between tasks (dataflow) in a workflow, e.g., specifying that tasks can be executed in parallel, or that a task needs to wait for data from other tasks
- exception handling, e.g., specifying what actions are necessary if a task fails or a workflow cannot be completed
- task duration, e.g., specifying initiation and completion time of a task
- priority attributes, e.g., specifying priorities for task scheduling

In rule- or constraint-based workflow specification languages, the workflow structure and dataflow are typically specified by defining *routing* rules or constraints. Routing is often classified as *conditional*, *rule-based*, or *parallel*. Conditional routing involves scheduling a task based on data values. For example, "if item.cost > 1000, then contact Manager." Rule-based routing is more powerful than conditional routing and can involve arbitrarily complex rules stated in a rule-based language. Parallel routing allows one task to branch into multiple others that can execute in parallel. A few languages also explicitly support task rendezvous.

Graphical user interfaces (GUIs) are provided for both graphical workflow specification and graphical task specification. Graphical workflow specification languages support the iconic representation of workflow tasks and the ability to sequence those tasks graphically by connecting arrows and decision icons among workflow tasks. Many WFMSs use graphical specification to automatically generate code or set up rules for a workflow implementation and execution.

GUIs for task specification support the creation of programming interfaces for tasks involving programs, and graphical interfaces for tasks involving humans.

Testing, analysis, and monitoring tools. Workflow testing tools simulate a workflow by allowing input of sample data and triggering events such as task completion, deadline expiration, and exceptions. Simulation is needed to uncover logic errors and get estimates of workflow completion times.

Workflow analysis tools are needed to predict possible bottlenecks in a workflow by analyzing the workflow specification. The analysis is done by taking into account workflow execution or simulation statistics. For example, analysis tools can gather statistics on workflow performance and suggest alterations to the workflow specification to improve efficiency. Some products supply simple testing and/or analysis tools, but typically they are inadequate.

Once a workflow is implemented, we need to monitor its progress, e.g., for checking the status of a workflow, or determining bottlenecks. WFMSs provide GUIs that can present different views of workflow execution, e.g., they illustrate which task or tasks are currently active, by whom they are performed, the task priorities, task deadlines, task durations, and task dependencies. Managers can use such monitoring tools to access workflow statistics such as task completion times, workloads, and user performance, as well as to generate reports and provide periodic summary of workflow executions.

Systems architecture and interoperability. Some commercial WFMSs have open client-server architectures and complete application programming interfaces (APIs) (i.e., such that everything that can be done through the user interface can also be done via an API).

WFMSs support exchange of information among users or systems via email or a shared (usually WFMS vendor proprietary) database (in Appendix B we use the term transport to refer to the way information exchange is implemented). Email supports human notification and databases are used to maintain shared documents. Administrative WFMSs are often based on email. Ad hoc and production WFMSs typically store information in a shared database. Many WFMSs provide a combination of these.

Many WFMSs support limited interoperability among some office applications that manipulate documents. For example, Lotus Notes uses Microsoft's OLE which provides a protocol for document interoperability.

Implementation support. Although GUIs and APIs contribute greatly to implementation support, there are other capabilities (i.e., with or without GUIs) that support ease of implementation, maintenance, and use. These include:

- *dynamic modification of workflow*—the ability to change task sequencing or introduce new tasks into an executing workflow

- *event signaling and notification*—the ability for programmers to raise events in one task and have another task "notice" that event and take action on it. This allows a loose coupling among tasks without embedding rules in task code. For example, a WFMS may support deadline management which notifies users when active tasks approach their deadline
- *user administration*—associate users to roles and support the management of these associations

Dynamic Workflow Modification, Event Signaling (Event-Action Triggers), and Role/User Administration are popular WFMS features. They are provided by all WFMSs compared in Appendix B.

Correctness and reliability. When multiple objects (e.g., databases, files, documents, devices) are accessed by a workflow execution, data consistency problems can arise either from concurrency, application failures, system failures, or network failures. These introduce the need for *concurrency control*, *recovery*, and *transaction coordination*. Most commercial WFMSs provide limited capabilities to deal with these problems, with a few notable exceptions such as FlowMark which keeps a log of performed actions and workflow states and supports automatic restart after failures [25].

In many situations concurrency control is essential when two or more users (or computer systems) can access the same data object. However, commercial WFMSs take widely different approaches to concurrency control, depending on perceived workflow requirements. For example, some WFMSs (e.g., XAIT's InConcert) support a form of check-in and check-out on data items such that users can "lock" data to preclude concurrent access by other users. Check-in and check-out is a primitive way to handle concurrency when compared to how DBMSs support concurrency, and may not ensure workflow consistency.

Other WFMSs (e.g., Lotus Notes) allow multiple users to retrieve the same data object concurrently. If each user decides to update that data object, new versions of the data item are created to be reconciled (merged) by human intervention. The rationale for this approach is the assumption that data object updates are rare. Thus, consistency can be handled by humans who review the data object versions and decide which one they want to keep. If this assumption is not met, this approach has potentially serious ramifications. Consider an example where multiple users retrieve hundreds if not thousands of data objects from a WFMS-controlled database, perform some automated procedure on them, and return those changes to the database. This may result in thousands of object versions all requiring human intervention to merge. Clearly, this poses a problem not only because of the time required to review these objects, but also the potential for conflicting information which cannot be correctly merged.

Still other WFMSs (e.g., Sietec's Staffware) use a pass-by-reference/pass-by-value approach for concurrency control. Data items (e.g., documents) that can be shared among multiple users are passed by reference, i.e., users access a centrally stored data item using a handle (pointer), possibly concurrently. This approach requires some form of concurrency control to provide an adequate level of non-interference among users.

Workflow recovery involves (i) how to undo completed or partially completed tasks that cannot be completed due to a failure, and (ii) how to undo a cancelled workflow. For example, consider a telephone service provisioning workflow that, among other things, updates a customer database and billing database, and allocates facilities for the customer. If a customer requests service and later cancels service before installation is completed, two options are possible: (i) allow the workflow to complete and execute a separate workflow that cancels service, or (ii) discontinue the workflow and undo completed tasks. The first option is simpler, since it does not involve undoing tasks. However: (i) compute power is wasted to finish processing a workflow known to be useless, (ii) human effort is wasted if facilities must be installed to support service that will soon be disconnected, (iii) a second workflow must be executed taking additional resources, and (iv) allocated facilities cannot be used to support the needs of other workflows.

The second option, i.e., stopping the workflow as soon as it is known to be useless, avoids these concerns. However, the ability to stop, or *abort*, a workflow in the middle of its execution requires additional support from the WFMS. In particular, the WFMS must maintain the state of each task, and must use these to reach a *consistent* state from which it can undo the effects of failed workflows.

Most commercial WFMSs rely on workflow designers for providing workflow specification to deal with reliability problems. For example, InConcert deals with workflow and task failures by allowing dynamic modification of workflows to specify tasks that perform compensation or alternative actions. As tasks and workflows become more automated, however, the speed at which business processes are performed and the volume of data being affected makes human-controlled recovery impractical. No commercial WFMSs we are aware of offers significant support for workflow recovery.

4.2. Limitations of workflow management systems

Despite their many features, WFMSs have a number of significant limitations. These include:

- lack of interoperability among WFMSs
- lack of support for interoperability among HAD systems or among WFMS and HAD systems
- inadequate performance for some business processes
- lack of support for correctness and reliability
- weak tool support for analysis, testing, and debugging workflows

Lack of interoperability among WFMSs. This is directly attributable to the lack of standards for WFMSs. As discussed in Section 3.2, the Work Flow Management Coalition, a standards body, was formed in 1993 to promote interoperability among WFMSs. Their standards address the areas of (i) APIs for consistent access to WFMS services/functions, (ii) specifications for formats and protocols between WFMSs themselves, and between WFMSs and applications, and (iii) workflow model interchange specifications to allow the interchange of workflow specification among multiple WFMSs. Most WFMS vendors are members of this coalition.

Lack of support for HAD system interoperability and integration. For workflows that access HAD information systems, full interoperability among HAD systems, WFMSs, and workflow implementations is important for the following reasons:

1. it simplifies workflow implementation, i.e., interoperability allows WFMSs to access HAD systems without requiring any HAD system-specific code,

2. it allows fast workflow implementation, i.e., workflow implementations that do not include HAD system-specific code can be developed faster that those that involve programming, and
3. it requires minimal workflow re-implementation to cope with changes in HAD system functionality, i.e., it requires no code changes in workflow implementations except re-specification of HAD system interfaces.

Some commercial WFMSs support limited interoperability among office applications meeting specific platform, interface, or operating system requirements. For example, some WFMSs (e.g., Lotus Notes) use Microsoft's OLE as a protocol for document interoperability. However, further interoperability requires WFMSs to take advantage of technology that complies with industry standards for interoperability, such as those developed by the Object Management Group (OMG).

Inadequate performance. Commercial WFMSs typically support no more than a few hundred workflows a day. Some processes require handling a larger number of workflows; perhaps a number comparable to the number of transactions TP systems are capable of handling. For example, telecommunications companies currently need to process ten thousand service provisioning workflows a day, including a few thousand service provisioning workflows per hour at peak hours. Commercial WFMSs are currently not capable of handling such workloads.

Lack of support for correctness and reliability in the presence of concurrency and failures. Workflow execution (like any execution of applications that access shared resources) must address the following three correctness concerns: the consistency of individual tasks, the consistency of individual workflows (i.e., the consistency of concurrent executions of tasks that belong to the same workflow), and the consistency of concurrent executions of tasks that belong to different workflows. Usually, the person who implements a task is responsible for ensuring that the task produces correct results if it is executed alone. It is also reasonable to assume that if correct tasks are executed one after the other in an order allowed by workflow specification rules, such an execution of a workflow preserves consistency by design. However, if tasks are executed concurrently with tasks of the same or other workflows, and the tasks share resources, their individual operations may interleave in such a way as to produce incorrect results. These are well-understood issues in database transaction processing.

The workflow reliability problem involves restoring consistency when a workflow terminates abnormally (e.g., due to a system failure, lack of available resources, or inability to achieve objectives). Completed tasks of a partially completed workflow may be undone or compensated. Alternatively, incomplete tasks of a partially completed workflow may need to be redone or contingency tasks need to be performed. Clearly, it is important to know which tasks have been completed, which are still active, which have not begun, and which tasks need to be undone or redone to restore consistency.

To deal with these problems, WFMSs rely on (i) workflow designers for providing specifications that include compensating tasks and actions, and (ii) task/workflow programmers for providing code for concurrency control and keeping logs. This solution is unrealistic, as most workflow designers/programmers are not skilled in concurrency control and recovery technology. Moreover, software to implement concurrency control and recovery mechanisms is complex. Finally, testing and debugging software with hard-coded correctness and reliability functions is time-consuming and error prone.

Weak tool support for analysis, testing, and debugging of workflow specifications and implementations. As discussed in Section 4.1, such tools are needed to estimate workflow specification and implementation efficiency, simulate workflow execution, and determine the source of workflow specification and implementation problems. The sophistication of such tools directly impacts rapid prototyping, and ease of workflow specification and implementation.

Overall evaluation. Some limitations of current WFMSs can be attributed to their youth, e.g., lack of standards and tools for conceptual modeling, testing, debugging, and analysis. Other limitations, however, are fundamental to the design of the WFMSs we have investigated, e.g., lack of support for interoperability, as well as correctness and reliability support. In the following section we discuss technologies that can be integrated with workflow management to address these limitations.

5. Key infrastructure technologies for workflow management

Efficient and reliable support for workflow implementation and execution requires a distributed computing environment that:

- supports integration and interoperability among loosely-coupled HAD legacy and new systems,
- supports workflow applications that require access to multiple HAD systems,
- ensures correctness and reliability of workflow applications in the presence of concurrency and failures, and
- supports evolution, replacement, and addition of workflow applications and systems as business processes change.

Commercial WFMS satisfy some of these requirements. In particular, commercial WFMS facilitate the process of specifying and implementing workflows by: (i) providing a constrained environment using a limited set of concepts to specify and implement workflows, and (ii) keeping workflow structure (i.e., the rules for sequencing of tasks) separated from the task implementation code. The latter allows changes in workflow structure without modifying the programs that implement the workflow tasks. Thus, WFMSs support efficient (re)design and (re)implementation of workflows as business needs change. In addition, WFMSs cope with changes in HAD systems and the tasks they support by re-implementing affected workflows.

To address the remaining requirements, two key infrastructure technologies, Distributed Object Management (DOM) and Customized Transaction Management (CTM), can be combined with the WFMS capabilities that commercial WFMS already provide. DOM, as partly exemplified by software complying with the OMG's Common Object Request Broker Architecture (CORBA) [35, 36], is one of several fast maturing distributed computing technologies and standards that are useful in providing HAD system interoperability. We believe than DOM is particularly useful for supporting workflow management for the following reasons:

- it supports the interoperability of HAD systems and applications implementing workflows,

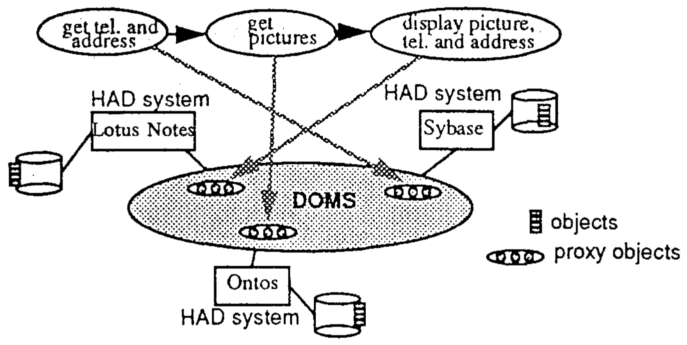

Figure 12. An example of a simple DOMS providing interoperability for ad hoc workflow.

- it copes with changes that result from replacement, migration and evolution of HAD systems, and
- it provides an object model that helps manage complexity and provides more transparency than other distributed computing technologies exemplified by DCE [BGHM93] (other infrastructure alternatives such as DCE may be useful in the short term until DOM technology becomes more mature).

DOM is discussed further in Section 5.1
CTM complements DOM by:

- providing the correctness and reliability each workflow requires,
- dealing with differences in HAD system-provided correctness and reliability guarantees, and
- coping with changes in application correctness and reliability requirements.

CTM is described in Section 5.2.

5.1. Distributed object management

DOM [30, 35, 36, 28] supports the interoperability and integration of component HAD systems by: (i) representing their data and functionality as objects, and (ii) allowing client applications (such as workflow implementation) to invoke behavior on server objects typically without regard to an object's location, data representation, or access language. In addition, DOM provides an object model that facilitates managing complexity by the use of abstraction, inheritance, and polymorphism. A *Distributed Object Management System* (DOMS) is a system that uses DOM technology.

In Fig. 12, we illustrate a workflow implementation using a DOMS. The DOMS provides interoperability among two HAD systems (Ontos^{TM} and Sybase^{TM} DBMSs) and a commercial WFMS (Lotus Notes^{TM}). In addition, the DOMS provides interoperability and reuse for workflow tasks that are represented as DOMS objects. The workflow in our example groups and presents multimedia information for the reviewers of a conference paper.

In this example, the Sybase database contains relations that include the name, address, telephone number and other textual data for the people in the mailing list. Ontos is an

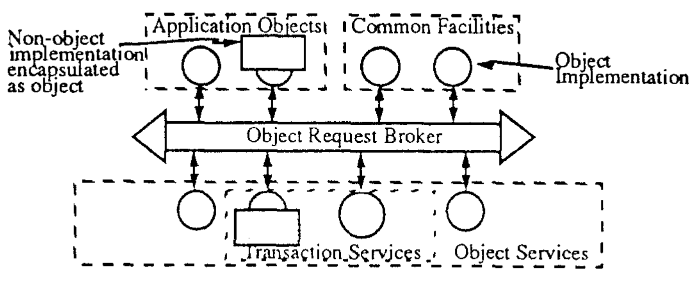

Figure 13. Object Management Architecture.

object DBMS that stores voice messages and pictures as objects. Both Sybase and Ontos behave as servers in the DOMS. Lotus Notes is a DOMS client that serves as an interface for the workflow users and stores workflow functionality that manipulates multimedia data. In the DOMS, proxy objects are defined to represent (i) the mailing list data (e.g., records, relations, objects, or an entire database) and functionality (e.g., query processing) provided by Sybase and Ontos, and (ii) the workflow tasks (to allow task reuse). Finally, additional objects could be defined in the DOMS to create composite objects representing composite tasks, or even workflows. When the workflow in our example is executed it invokes the task objects. The task objects invoke the multimedia objects (i.e., issues queries over the multimedia data), and in response, the DOMS transparently retrieves the distributed data and returns these to Lotus Notes.

DOMS support for full interoperability simplifies the building of workflow implementations and the workflows themselves (e.g., by supporting task reuse and composition).

5.1.1. Technology status. To standardize DOMS architecture and services, software vendors formed OMG. OMG's Object Management Architecture [36] consists of a DOMS component called Object Request Broker (ORB) [35] and the following three classifications of objects: Applications, Object Services, and Common Facilities [36] (illustrated in Fig. 13). The Common Object Request Broker Architecture (CORBA) [35] is the DOMS architecture developed by OMG. An ORB is a CORBA-compliant DOMS. *Applications* are essentially the clients of the ORB environment. *Object Services* is a collection of interfaces and objects that provide basic functions for using and implementing objects (e.g., transaction services). *Common Facilities* is a collection of non-standard services that provide general purpose capabilities useful in many applications. Applications access server objects (i.e., services and common facilities) via the ORB. Note that server objects themselves may be clients of other server objects.

There are several commercial products that comply with CORBA. Commercial ORBs include Iona's Orbix, DEC's ObjectBroker, IBM's DSOM, HP's DOMF, and Sun's DOE.

5.2. *Customized transaction management (CTM)*

CTM can support specific correctness, reliability, and functionality requirements for each workflow application. To define CTM precisely, we first characterize the transactional capabilities of the HAD systems workflow applications need to access assuming that a DOMS is used to integrate HAD systems and WFM.

From the perspective of transaction management, objects integrated by a DOMS fall into one of the following categories:

- *Transactional objects* representing data and functionality in HAD systems that support transactions. Any object that implements its own *transaction management mechanism* (TMM) is included here. For example, this category includes DBMSs as well as those file systems and programming language systems, e.g., Argus [26], that provide transaction support similar to that provided by a DBMS. We refer to DBMSs and all other systems in this category as *Local Transaction Management Systems* (LTMSs).
- *Transactionless objects* that represent data and functionality in HAD systems that do not support transactions. Local systems in this category may be multi-threaded (e.g., file systems) or single-threaded (e.g., word processors, spreadsheets, simple programs). Multi-threaded local systems may provide some built-in concurrency control. Commercial WFMSs belong in this category.

For example, in the DOMS and HAD systems in Fig. 12, the Ontos and Sybase DBMS are LTMSs. The databases, relations, tuples, and objects maintained by the DBMSs, and/or the DBMSs themselves, are transactional objects. Lotus Notes is a transactionless local system. Lotus Notes data and functionality, and the workflow implementation are transactionless objects.

To ensure the correctness and reliability of workflows accessing multiple transactional and/or transactionless objects, each workflow application must be associated with a *transaction model* that defines the correctness and reliability required by the workflow. DBMSs and TP monitors provide the ACID transaction model as a default to all applications using their transaction management services. This basic transaction model requires the enforcement of *all* ACID properties (i.e., atomicity, consistency, isolation, and durability), e.g., allows applications to interleave as long as they produce results equivalent to a non-interleaved execution. However, the ACID transaction model is not appropriate for many workflow applications. For example, since the ACID transaction model enforces task isolation it does not permit task cooperation. Therefore, it is too restrictive for workflows that require task interaction.

CTM is the technology that satisfies these requirements. In particular, CTM supports the definition and enforcement of ETMs that are:

1. *application-specific*: support workflow applications that have specialized requirements for correctness, reliability, and functionality (e.g., require task cooperation)
2. *user-defined*: define constraints that are not built into the code of the TMM that enforces them, since application requirements and object-provided guarantees may change
3. *augmented*: require correctness and reliability capabilities that may not be supported by an object (and the corresponding HAD system)
4. *multi-system*: support applications that require access to objects maintained by multiple HAD systems

Since these extend the ACID transaction model typically supported by DBMSs and TP monitors, we refer to any transaction model that has at least one of these properties as an *extended transaction model* (ETM).

Application-specific ETMs define the correctness and reliability requirements of applications implementing workflows. *Transactional workflows* are workflows supported by an ETM that defines workflow correctness and reliability criteria. In particular, mapping workflows to extended transactions involves [15]:

1. mapping workflow tasks to constituent transactions of an extended transaction supported by an ETM,
2. mapping workflow structure to an extended transaction structure supported by the same ETM as (1), and
3. ensuring that workflow execution obeys the correctness criterion defined by the ETM.

Typically, workflow requirements are so diverse that no single ETM is sufficient to meet the needs of all workflows.

The need for introducing ETMs to extend the traditional (ACID) transaction model typically supported by DBMSs to allow additional application functionality (e.g., permit task collaboration and coordination as it is required by ad hoc workflows) and improve throughput (e.g., reduce transaction blocking and abortion caused by transaction synchronization) has been recognized for some time. Several ETMs have been proposed (refer to [10, 16] for frameworks for defining and comparing ETMs, [13, 12] for several representative ETMs, [43] for a representative model and specification that support application specific transaction properties, and [7, 5, 15, 20, 39] for views on relationships between workflows and ETMs). However, many of these ACID transaction model extensions resulted in application-specific ETMs offering adequate correctness guarantees in a particular application, but not ensuring correctness in others. Furthermore, an ETM may impose restrictions that are unacceptable in one application, yet required by another. If no existing ETM satisfies the requirements of an application, a new ETM must be defined to do so. We give an example in the following section.

ETMs must be user-defined (not built-in) to effectively support: (i) a variety of transactional workflows with diverse and possibly conflicting ETMs, and (ii) ETM changes that may result from the evolution of a workflow and experience with its correctness and reliability requirements (e.g., correctness constraints may be relaxed or imposed as we gain more experience with an application). Traditionally, a system developed for a particular application was designed to provide a built-in transaction model supporting application requirements. In such systems, selecting an appropriate ETM was the responsibility of the system designer. A DOMS or WFMS that supports a single built-in ETM cannot satisfy the requirements of many diverse transactional workflows and cannot take advantage of application experience in the (re)definition of the workflow ETM.

Defining an augmented ETM involves comparing the ETM correctness requirements with the correctness guarantees provided by the HAD systems to determine whether the ETM can be enforced by the local HAD systems. If the local HAD system cannot enforce an ETM, CTM provides additional correctness and reliability guarantees that complement those provided by the HAD system and together meet ETM requirements. For example, consider the DOMS in Fig. 12 and a transactional workflow that requires correctness and reliability guarantees similar to those provided by ACID transactions. To satisfy the workflow requirements CTM must provide ACID transactions over transactionless objects in Fig. 12.

The problem of supporting multi-system transactions has been partially investigated in research on transaction management in multidatabase systems [19, 8]. However, unlike multidatabase transactions, transactional workflows may not consist of ACID transactions and may have more than two levels of transaction/task nesting [43, 39, 11].

5.2.1. Specification of transactional workflows. Specification of transactional workflows involves the specification of the ETMs that define their correctness criteria and

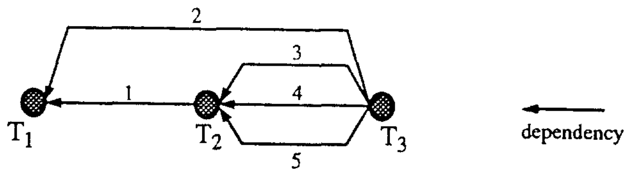

Figure 14. Dependencies that result from the execution structure of transaction T.

structure. ETM specification is based on the observation that extended transactions consist of a set of *constituent transactions* (corresponding to the workflow tasks) and a set of *transaction dependencies* between them (corresponding to the workflow structure and correctness criterion). If a transactional workflow involves nesting, the constituent transactions of a transactional workflow may be extended transactions themselves. Each transactional workflow T has the following two kinds of transaction dependencies:

- *Intra-workflow transaction dependencies* that define the relationships between the constituent transactions of T.
- *Inter-workflow transaction dependencies* that define the relationships between T and all other transactional workflows.

To illustrate transaction dependencies, consider again the "New Service Provisioning" workflow illustrated in Fig. 2. To simplify the discussion, suppose that T is the subset of the "New Service Provisioning" workflow that includes only tasks T_1, T_2, and T_3. In particular, assume that T is executed when a telephone company customer requests telephone service installation. Task T_1 performs a transaction that registers billing information in the customer database. Tasks T_2 and T_3 execute transactions that perform two alternative line provisioning functions. Only one of the provisioning tasks should be allowed to complete as either will result in a completed circuit, i.e., a set of lines and equipment that connects the customer to a telephone network. T_2 attempts to provide a connection by using existing facilities such as lines and slots in switches. If T_2 succeeds, the cost of provisioning is minimal, i.e., the requested connection is established by allocating existing resources. However, a successful completion of this activity may not be possible if the facilities are not in place or if the capacity of existing facilities is exhausted. T_3 achieves the same objectives as T_2 but involves physical installation of new facilities. Thus, T_3 has a higher cost than T_2. Since T_3 is needed only if T_2 fails, T_2 and T_3 are contingency transactions.

The first category of intra-workflow transaction dependencies are *transaction state dependencies*. Dependencies in this category are conditions on transaction state that define the *execution structure* of transactional workflows. Defining intra-workflow transaction state dependencies involves using transaction semantics (e.g., Begin, Abort, Commit) to specify the ordering of workflow tasks. Such specifications that are more precise than those allowed by the workflow specification languages WFMS typically provide. Figure 14 depicts the dependencies that define the execution structure of T assuming that T_2 and T_3 cannot begin before T_1 commits and that T_2 and T_3 can execute concurrently.

The constituent transactions of T have the following transaction state dependencies:

1. T_2 cannot begin *before* T_1 commits
2. T_3 cannot begin *before* T_1 commits
3. T_3 cannot commit *before* T_2 aborts

4. T_3 *must* abort if T_2 commits
5. T_3 cannot begin *after* T_2 has committed

The second category of intra-workflow transaction dependencies are *correctness dependencies* that specify which concurrent executions of transactional workflows preserve consistency and produce correct results, thereby defining a *correctness criterion.* Correctness dependencies include:

- *Serialization dependencies* specify whether operations performed by a set of transactional workflows must be serializable [9].
- *Visibility dependencies* define whether operations performed by a set of transactional workflows must be recoverable, cascadeless, strict, or rigorous [9, 8].
- *Cooperation dependencies* define whether a set of transactional workflows may perform specific operations on specific objects without restrictions.
- *Temporal dependencies* that specify a set of transactional workflows must perform operations in a particular temporal order [18].

The following dependencies are sufficient to ensure correctness of the transactional workflow T in our example:

- *inter-workflow transaction serialization dependencies*: The operations performed by T and the operations performed by all committed transactional workflows must be serializable. Note that this dependency does not require the constituent transactions of T to appear atomic to each other.
- *intra-workflow transaction cooperation dependencies*: The two contingency transactions T_2 and T_3 need not appear atomic to each other.

The cooperative dependencies between T_2 and T_3 may result in a situation in which both transactions are able to construct complete circuits using the same line(s) and/or equipment. Since circuits require exclusive access to their lines, this will be unacceptable for many other provisioning workflows. However, the semantics of this particular telecommunications application permits any number of alternative transactions like T_2 and T_3, as long as only one of them is allowed to commit and use the lines.

5.2.2. Technology status and research towards supporting CTM. Commercial DBMSs, TP monitors, and multidatabase systems currently support only a small set of generic ETMs (e.g., flat, nested, transaction consistency, cursor stability, uncontrolled reads). The CORBA transaction service [37] is essentially a TP monitor. Thus, it is subject to the same ETM limitations as the other TP monitors. Except for multidatabase transaction support, none of these commercial products provides application-specific or user-defined ETMs.

The CTM concept, and research for developing technology to provide CTM, are relatively new [17, 5]. The *Transaction Specification and Management Environment* (TSME) [17] is an example of a research effort for developing CTM technology. To support the specification of ETMs, and the configuration of corresponding transaction management mechanisms, the TSME provides a transaction dependency specification facility (TDSF) and a corresponding programmable transaction management mechanism (PTMM). The TDSF accepts specifications of ETMs expressed in terms of dependencies between the kinds of transactions allowed by each ETM (as described in Section 5.2.1). The programmable TMM supports the implementation of ETMs, i.e., provides an environment in which to execute transactional workflows and ensures the preservation of their dependencies/ETMs.The TSME framework for ETM specification is described in [16]. The TSME architecture and programmable TMM are discussed further in [17]. The ASSET system [5] is another research effort for supporting application-specific ETMS. ASSET provides a PTMM facility but does not deal with object/HAD system autonomy.

6. Conclusion

Workflow is an intuitive and powerful paradigm for capturing business processes, reasoning about them, and using process specifications to produce corresponding implementations that are supported by the information systems. However, the scope of solutions provided by commercial WFMS is limited, e.g., they only support document/form/image-centered processes, CSCW and office automation applications. Furthermore, many WFMS products do not support workflows that need functionality and data in legacy HAD systems, do not address efficient HAD system integration and interoperability, and do not ensure the correctness and reliability of workflow execution in the presence of concurrency and failures.

Workflow research is also fragmented across multiple disciplines, such as CSCW, computer-human interaction, imaging, and databases. The lack of interdisciplinary workflow research has hindered establishing a common understanding or an understanding of the different perspectives. For example, database researchers view workflows as information processes and do not consider the human aspects of the business process implementation. On the other hand, CSCW researchers ignore the role and importance of information systems. Since the vision of WFMSs is to support business processes that span entire organizations or multiple organizations, and involve tasks that are performed by humans as well as information systems, workflow requires technology involving the integration of technology from all these research areas.

In this paper, we discussed the requirements a computing environment supporting workflows needs to satisfy. Furthermore, we described the reasons we believe WFM, DOM, and CTM technologies should be integrated, and gave an overview of the technology status and research in progress in these areas. Figure 15 depicts the dependencies between WFM, CTM, and DOM in providing a computing infrastructure that supports fundamental workflow management requirements.

Using commercial software available today to develop such a workflow infrastructure may involve an open (extensible) WFMS, a commercial ORB, and a TP monitor. CORBA products and XT/XA compliant TP monitors are the current state of art for DOM and multi-

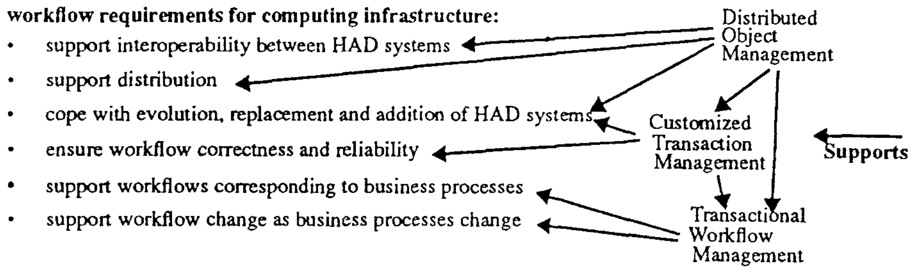
workflow requirements for computing infrastructure:
• support interoperability between HAD systems
• support distribution
• cope with evolution, replacement and addition of HAD systems
• ensure workflow correctness and reliability
• support workflows corresponding to business processes
• support workflow change as business processes change
Distributed Object Management
Customized Transaction Management
Transactional Workflow Management
Supports

Figure 15. Integration of DOM, CTM, and WFM to support transactional workflows.

system transaction processing, respectively. OMG has recently integrated TP monitor technology by standardizing the OMG transactions services. Unfortunately, most WFMSs are not CORBA compliant. TP monitors do not provide CTM functionality required by many workflows, since TP monitors provide only a few built-in transaction models. Therefore, supporting organization-wide workflows today requires the development of software adaptors for bridging CORBA and commercial WFMS architectures and services, and relying on software that operates outside the control of a TP monitor for CTM functionality. Although many difficult problems must be solved before the workflow management technology becomes mature, the workflow concept is a compelling vision for research and commercial development.

Appendix A: Commercial product and company list

Action Workflow	Action Technologies 1301 Marina Village Pkwy., Suite 100, Almeda, CA 94501, 800-WORKFLO. 510-521-6190
Amadeus	Amadeus Software Research Inc. 12 Owen Court, Irvine, CA 92715, (714) 725-6400, email: amadeus-info@amadeus.com
Beyond Mail	Beyond, Inc. 800-845-8511, 617-229-0006, fax: 617-229-1114
COHESION Team/SEE	Digital Equipment Corp. 800-344-4825, 508-493-5111, fax: 508-493-8780
Design/IDEF	Meta Software 125 Cambridge Park Road, Cambridge, MA 02140, 617-576-6920
EPIC	Computron Technologies
Extend + BPR	Imagine That 6830 Via Del Oro, Suite 230, San Jose, CA 95119, 408-365-0305, fax: 408-629-1251
Financial Stream	Dun & Bradstreet Software 3445 Peachtree Rd. NE, Atlanta, GA 30326-1276, [Accounting related], 404-239-2000
FloMark	IBM 800-426-2968, 800-426-3333, 914-765-1900
FloWare and MapBuilder	Recognition International 1310 Chesapeake Terrace, Sunnyvale, CA 94089, 800-999-5910, 408-743-4300, fax: 408-747-1245
Flowman	Logical Software Solutions 7701 Greenbelt Rd., Suite 207, Greenbelt, MD 20770, 301-474-0285, fax: 301-474-0386
FormFlow	Delrina Corp., 416-441-3676, fax: 416-441-0333
Higgins	Enable Software 800-888-0684, 518-877-8600
ImageMover	IMC, Inc.
InConcert	XSoft (a division of Xerox Corp.) 3400 Hillview Ave., Palo Alto, CA 94303, 800-428-2995, 415-424-0111, fax: 415-813-7162
Keyfile	Keyfile Corp. 22 Cotton Rd., Nashua, NH 03063, 603-883-3800, fax: 603-889-9259
LinkWorks	Digital Equipment Corp., 110 Spitbrook Rd., Nashua, NH 03062, (603) 881-146, fax: 603-881-2550
Mate	Advanced Development Methods, Inc. 617-861-7848

Electronic Forms Designer	Microsoft Corp. 800-426-9400, 206-882-8080
MSP/PM	Software Design & Analysis Inc. 1113 Spruce Street, Boulder, CO 80302, 303-444-9100. fax: 303-938-5005, email: riddle@sda.com
N-Dimensions	Computron Technologies [Accounting related]
Navigator 2000/Workflow	I. Levy and Associates 1600 Des Peres Rd., Suite 300, St. Louis, MO 63131, 314-822-0810, fax: 314-822-0309
Notes	Lotus Development Corporation, Cambridge, MA, 617-577-8500, 800-346-1305
ObjectFlow	Digital Equipment Corporation, contact: Daryl Rosen, (603) 884-0882
OmniDesk	Sigma Imaging Systems, Inc. 622 Third Ave., New York, NY 10017, 212-476-3000, fax: 212-986-0175
Open/workflow	Wang Labs One Industrial Ave., Lowell, MA 01851, 800-225-0654, 508-459-5000, fax: 508-967-0828
Optika	Optika Imaging Systems 5755 Mark Dabling Blvd., Suite 100, Colorado Springs, CO 80919, 719-548-9800, fax: 719-531-7915
PCMS	SQL Software Ltd Latton Bush Centre, Southern Way, Harlow, Essex CM18 7BL, UNITED KINGDOM, +44 (0) 279 451724 In the U.S., contact: SQL Software Inc. 8000 Towers Crescent Drive, Suite 1350, Vienna, VA 22182, (703) 760-7895, fax: 703-760-7899
PCTE	Emeraude 152 Bureaux de la Colline, 92213 Saint Cloud Cedex, FRANCE, +33 1 49 11 72 68, fax: +33 1 49 11 72 40
ProcessIt	AT&T Global Information Solutions 1700 S. Patterson Blvd. Dayton, OH 45479, 800-225-5627, 513-445-5000, fax: 513-445-4184
Process Weaver	Cap Gemini Innovation 7, chemin du Vieux Chene, 38240 Meylan, FRANCE, +33 76 76 47 47, fax: +33 76 76 47 48 In the U.S., contact: Cap Gemini America, Software Engineering Productivity Practice, (212) 944-6464
Processwise	ICL Wenlock Way, West Gorton, Manchester, M12 5DR, UK 061 223 1301
ProEDITOR	International Software Systems, Inc. 9430 Research Blvd., Echelon IV, Suite 250, Austin, TX 78759 512-338-5700, fax: 512-338-5757
Redwood	Walker Interactive Systems 303 Second St., Ste. 3 North, San Francisco, CA 94107 [Accounting related] 415-49597711, fax: 415-543-6338
Staffware	Staffware Corp.
SynerVision	Hewlett Packard Company U.S.: 800-63797740 Canada: Mississaugua, Ontario, 416-678-9430 Japan: Tokyo, 03 5371 1351 Latin America: Lomas de Chapultepec, Mexico, 525-202-0155 Australia/New Zeland: Blackburn, Australia, 03 895 2895 Asia Pacific: Hong Kon, 852-848-7777 Europe/Africa/Middle East: Geneva, Switzerland, 057-31-21-11

TeamFlow	CFM, Inc.
TeamFlow—Client	ICL 9801 Muirlands Blvd., Irvine, CA 92713, 714-855-5500, fax: 7149-458-6257
Teamlinks Information Manager	Digital Equipment Corp. 800-344-4825, 508-493-5111, fax: 508-493-8780
TeamSync	Global Stream Corp. 800-685-7858, 206-858-7858
Team Talk	Trax Softworks, Inc. 800-FOR-TRAX, 310-649-5800
TASC-Flow, TRAC-FlowSlim, TASC-Control	TASC 55 Walkers Brook Dr., Reading, MA 01867 617-942-2000, fax: 617-942-7100
ViewStar	ViewStar Corp. 1101 Marina Village Pkwy., Alemeda, CA 94501 510-337-2000, fax: 510-337-2222
Visual Workflo	FileNet Corp. 3565 Harbor Blvd., Costa Mesa, CA 92626 800-FILENET, 714-966-3400, fax: 714-966-3490
WorkFast	ImageFast Software Systems 7926 Jones Branch Dr., Suite 260, McLean, VA 22102 800-899-6665, 703-893-1934, fax: 703-893-7499
WorkFlow Analyzer	Meta Software Corp. 125 Cambridge Park Dr., Cambridge, MA 02140, 800- 227-4106, 617-576-6920, fax: 617-661-2008
Workflow Manager	IBS 12626 Higg Bluff Drive, San Diego, CA 92130 800-346-02884, 619-772-273, fax: 619-792-5199
Workflow Manager	Intergraph 800-346-7920, 610-7100, fax: 610-293-7117
Workflow Management System	Cimage Corp. 3885 Research Park Dr., Ann Arbor, MI 48108, 800-241-4367, 313-761-6550, fax: 313-761-6551
Workgroup Templates	Microsoft Corp. 800-426-9400, 206-882-8080
WorkMAN	Reach Software Corporation 872 Hermosa Drive, Sunnyvale, CA 94086 800- MAILFLO, 408-733-8685, fax: 408-733-9265
WorkManager, WorkRouter	Hewlett-Packard Co. 300 Hanover St., Palo Alto, CA 94304 800-752-0900, 800-637-7740, fax: 800-333-1917
X Workflow	Olivetti 22425 E. Appleway Ave., Liberty Lake, WA 99019 800-633-9909, 509-927-5600, fax: 509-927-5700

Appendix B: A comparison of five commercial workflow management systems

Next we compare five WFMSs. The comparison matrix reflects information available from early 1994 product literature and conversations with sales and technical support staff from the WFMSs' respective vendors.

Feature\Product	InConcert	Staffware	FloWare	Notes	ActionWorkflow
Trade press classification	Production	Administrative	Production administrative planned: Ad hoc	Ad hoc	Production
Model	Jobs, tasks, roles, users	Case, procedure, roles, users	map, submap, activity, couriers	Tasks, jobs, forms, roles, users	Map, loop, act, roles, users
Client/server	Yes	Yes	Yes	Yes	Yes
Integration and launch of office applications	Yes	Yes	Yes	Yes	Yes (via Lotus notes)
Scripting language	Yes (adminis-trative func-tions only)	Yes	No (GUI only)	No	Yes
API languages	C, C++	No	C, XDP AD 4GL	C, C++	C, C++
Transport mechanism	RDBMS	Email	RDBMS	Proprietary DB	Email
Forms creation environment	No (planned integration with notes)	Yes	No	Yes	Yes
Dynamic workflow modification	Yes	Yes	Yes	Yes	Yes
Event signaling (event-action triggers)	Yes	Yes	Yes	Yes	Yes
Role/user administration	Yes	Yes	Yes	Yes	Yes
Routing	Conditional, parallel, rule-based	Conditional, parallel, rule-based	Conditional, parallel, rule-based	Conditional, parallel, rule-based	Conditional, parallel, rule-based
Testing or analysis tools	Yes	Yes	Yes	No	Yes
Graphical workflow definition/manipulation	Yes	Yes	Yes	Yes	Yes
Alarm/deadline/priority	Yes	Yes	Yes	Yes	Yes
Process tracking	Yes	Yes	Yes	Yes	Yes
Concurrency control	Yes (Check-in/ check-out)	Yes (Pass by reference/ value)	No	No (Replication/ versioning)	Yes (Primitive locking at the act level)
Support for recovery	No	No	No	No	No
Transaction coordination (e.g., 2PC)	No	No	No	No	No
Security	Auditability	Auditability electronic signatures authentication	Auditability authentication	Authentication encryption authentication	Auditability authentication data access Control
Reporting capabilities	Yes	Yes	Yes	Yes	Yes
Keyword search and workflow querying	Yes	Yes	Yes	Yes	Yes
OMG CORBA compliant	No	No	No	No	No

Acknowledgments

We thank Gail Mitchell, Sandy Heiler, and Frank Manola for their wonderful comments and suggestions.

Notes

1. Workflow implementations can *control* human tasks by directing humans to perform tasks as specified by the workflow. Workflow implementations can also *coordinate* human tasks in that they facilitate the exchange or coordination of business process information used by humans.

References

1. R. Alsop, "Workflow Automation Integration Requires A Large Technology Toolkit and A Structured Approach," Computer Technology Review, 1994.
2. M. Ansari, L. Ness, M. Rusinkiewicz, and A. Sheth, "Using Flexible Transactions to Support Multisystem Telecommunication Applications," Proceedings of the 18th Intl. Conf. on VLDB, August 1992.
3. Action Workflow System product literature. Action Technologies Inc., 1993.
4. D. Black, "Workflow Software: A Layman's Handbook, Part I," INFORM, April 1994.
5. A. Billiris, D. Gehani, H. Jagadish, and K. Ramamritham, "ASSET: A System for Supporting Extended Transactions," Proceedings of SIGMOD, 1994.
6. M. Bever, K. Geihs, L. Heuser, M. Muhlhauser, and A. Schill, "Distributed Systems, OSF DCE, and Beyond," *Proceedings of the DCE—The OSF Distributed Computing Environment, International DCE Workshop*, Karlsruhe, 1993.
7. Y. Breitbart, A. Deacon, H.-J. Schek, A. Sheth, and G. Weikum, "Merging Application-centric and Data-centric Approaches to Support Transaction-oriented Multi-system Workflows," in SIGMOD Record, 22(3), September 1993.
8. Y. Breitbart, D. Georgakopoulos, M. Rusinkiewicz, and A. Silberschatz, "On Rigorous Transaction Scheduling," *IEEE Trans. on Software Engineering*, 17(9), September 1991.
9. P.A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.
10. P. Chrysanthis and K. Ramamritham, "A Formalism for Extended Transaction Models," *Proceedings of the 17th VLDB Conference*, 1991.
11. W. Eckerson, "Case Study: The Role of IS in Reengineering," Open Information Systems, Patricia Seybold Group, Vol. 9, No. 2, February 1994.
12. A. Elmagarmid, Y. Leu, W. Litwin, and M. Rusinkiewicz, "A Multidatabase Transaction Model for InterBase," *Proceedings of the 16th International Conference on VLDB,* 1990.
13. *Database Transaction Models for Advanced Applications*, A. Elmagarmid (ed.), Morgan-Kaufmann, 1992.
14. C. Frye, "Move to Workflow Provokes Business Process Scrutiny," Software Magazine, April 1994.
15. D. Georgakopoulos, et al., "An Extended Transaction Environment for Workflows in Distributed Object Computing," in [20].
16. D. Georgakopoulos, and M. Hornick, "A Framework for Enforceable Specification of Extended Transaction Models and Transactional Workflows," International Journal of Intelligent and Cooperative Information Systems, World Scientific, September 1994.
17. D. Georgakopoulos, M. Hornick, Piotr Krychniak, and F. Manola, "Specification and Management of Extended Transactions in a Programmable Transaction Environment," Proceedings of 10th International Conference on Data Engineering, Houston, TX, February 1994.
18. D. Georgakopoulos, M. Rusinkiewicz, and W. Litwin, "Chronological Scheduling of Transactions with Temporal Dependencies," *VLDB Journal*, January 1994.
19. D. Georgakopoulos, M. Rusinkiewicz, and A. Sheth, "Using Ticket-based Methods to Enforce the Serializability of Multidatabase Transactions," *IEEE Trans. on Data and Knowledge Engineering*, February, 1994.

20. "Special Issue on Workflow and Extended Transaction Systems," M. Hsu (ed.), *Bulletin of the Technical Committee on Data Engineering* (*IEEE Computer Society*), 16(2), June 1993.
21. M. Hsu and C. Kleissner, "ObjectFlow: Towards a Process Management Infrastructure," submitted for publication.
22. I. Jacobson, *Object-Oriented Software Engineering—A Use Case Driven Approach*, ACM Press, Addison-Wesley, 1992.
23. P. Korzeniowski, "Workflow Software Automates Processes," Software Magazine, February 1993.
24. N. Krishnakumar and A. Sheth, "Managing Heterogeneous Multi-system Tasks to Support Enterprise-wide Operations," in this issue.
25. F. Leymann and D. Roller, "Business Process Management with FlowMark," *Proceedings of IEEE Compcon*, March 1994.
26. B. Liskov, "Distributed Programming in Argus," *Communications of ACM*, 31(3), March 1988.
27. S. McCready, "There is more than one kind of Work-flow Software," Computerworld, November 2 1992.
28. F. Manola, "MetaObject Protocol Concepts for a "RISC" Object Model," GTE Laboratories Incorporated, TR-0244-12-93-165, Dec. 1993.
29. R. Marshak, "Software to Support BPR — The value of Capturing Process Definitions," Workgroup Computing Report, Patricia Seybold Group, Vol. 17, No. 7, July 1994.
30. F. Manola, S. Heiler, D. Georgakopoulos, M. Hornick, and M. Brodie, "Distributed Object Management," *International Journal of Intelligent and Cooperative Information Systems*, 1(1), March 1992.
31. D. McCarthy and S. Sarin, "Workflow and Transactions in InConcert," Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society, Vol. 16, No. 2, June 1993.
32. R. Medina-Mora, T. Winograd, and R. Flores, "ActionWorkflow as the Enterprise Integration Technology," Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society, Vol. 16, No. 2, June 1993.
33. R. Medina-Mora, H. Wong, and P. Flores, "The ActionWorkflow Approach to Workflow Management," Proceedings of the 4th Conference on Computer-Supported Cooperative Work, June 1992.
34. "Building Networks: New York Telephone rethinks work to regain lost customers," Scientific American, November 1992.
35. "The Common Object Request Broker: Architecture and Specification," Pub. Object Management Group and X/Open, OMG Document Number 91.12.1, Rev. 1.1, 1991.
36. "Object Management Architecture Guide," OMG TC Document 92.11.1, 1992.
37. "Object Transaction Service," OMG TC Document 94.8.4.
38. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall, 1991.
39. M. Rusinkiewicz and A. Sheth, "Specification and Execution of Transactional Workflows," In Modern Database Systems: The Object Model, Interoperability, and Beyond, W. Kim, Ed., ACM Press, 1994.
40. T. Smith, "The Future of Work flow Software," INFORM, April 1993.
41. T. Schael and B. Zeller, "Design Principles for Cooperative Office Support Systems in Distributed Process Management," in *Support Functionality in the Office Environment*, A. Verrijn-Stuart (ed), North Holland, 1991.
42. T. Winograd and R. Flores, *Understanding Computers and Cognition*, Addison-Wesley, 1987.
43. H. Watcher and A. Reuter, "The ConTract Model," Chapter 7, In [13], 1992.