What do Advanced Transaction Models Have to Offer for Workflows ?

Devashish Worah and Amit Sheth
Large Scale Distributed Information Systems Lab
The University of Georgia
Athens, GA 30602-7404
email: {*worah, amit*}*@cs.uga.edu*
URL: *http://LSDIS.cs.uga.edu*

Abstract

Workflow management systems are finding wide applicability in small and large organizational settings. In this paper, we briefly review four large-scale applications to gauge their modeling and run-time requirements. Advanced transaction models (ATM) focus on maintaining data consistency and have provided solutions to many problems such as correctness, consistency, and reliability in transaction processing and database management environments. While such concepts have yet to be solved in the domain of workflow systems, database researchers have proposed to use, or attempted to use ATMs to model workflows. In this paper, we argue that workflow requirements in large-scale enterprise-wide applications involving heterogeneous and distributed environments far exceed the modeling and functionality support provided by ATMs, and suggest that an ATM is unlikely to provide a primary basis for workflow modeling and subsequent management. We have also presented various connotations of the term *transaction* that exist in the real-world organizational processes. Finally, we point out the need for looking beyond ATMs and using a multi-disciplinary approach for modeling large-scale workflow applications of the future.

Keywords: Advanced transaction models, large-scale workflow applications, transactions, workflows.

1 Introduction

There has been a growing acceptance of workflow technology in numerous application domains such as telecommunications, software engineering, manufacturing, production, finance and banking, health care, shipping and office automation [GHS95, SKM+96, PLS+96, BSR96, JAD+94,

*This research was partially funded by a cooperative agreement between the National Institute of Standards and Technology Advanced Technology Program (under the HIIT contract, number 70NANB5H1011) and the Healthcare Open Systems and Trials, Inc.

TV95, WF94, Smi93, PPVW96]. Workflow Management Systems (WFMS) are being used in inter- and intra-enterprise environments to re-engineer, streamline, automate, and track organizational processes involving humans and automated information systems.

Inspite of the extensive research and proliferation of commercial *workflow* products, workflow technology is relatively immature to be able to address the myriad problems associated with real-world applications. The current state of the art is dictated by the commercial market which is focused toward providing automation within the office environment with emphasis on coordinating human activities, and facilitating document routing, imaging, and reporting. However, the requirements for workflows in large-scale multi-system applications executing in heterogeneous, autonomous, distributed (HAD) environments involving multiple communication paradigms, humans and legacy application systems far exceeds the capabilities provided by products today [She95].

Some of the apparent weaknesses of workflow models that need to be addressed by the workflow community include the lack of a clear theoretical basis, undefined correctness criteria, limited support for consistency of concurrent workflows, lack of interoperability, scalability and availability, and lack of support for reliability in the presence of failures and exceptions [BDSS93, JNRS93, GHS95, MAGK95, AS96, KR96]. Emerging and maturing technologies such as CORBA [OMG95a], OLE/ActiveX, Web, Lotus Notes [Cor96] and Java, increasing need for electronic commerce using electronic data interchange (EDI) standards (e.g., ANSI X.12 and HL7), additional organizational requirements (e.g., security, authentication), need for integrated collaboration (not just coordination) support, increasing use of heterogeneous multimedia data, and support for dynamic workflows to respond to the fast changing environment (e.g., defense planning), or for supporting today's dynamic and virtual enterprises, have led to additional challenges that should be met by successful workflow-enabled solution in the future.

Workflow technology has emerged as a multi-disciplinary field with significant contributions from the areas of software engineering, software process management, database management, and distributed systems [SGJ+96]. In spite of the standardization efforts of the Workflow Management Coalition [Coa94], a consensus on many broader aspects have not yet been achieved

Work in the areas of transaction processing [GR93] and database systems, and many (but not all) efforts related to the advanced transaction models (ATM) [Elm92, CR90, CR92], are based on a strong theoretical basis and have proposed or documented solutions (although many of which have yet to be implemented) to problems such as correctness, consistency, and recovery when the constituent tasks are transactional or the processing entities provide a transactional interface. There exists a strong school of thought, primarily comprised of researchers from the database community, which views a workflow model as an extension of ATMs [GH94, GHKM94, CD96, BDG+94, Wei93, WR91]. However, it has also been observed [BDSS93, AAA+96] that ATMs have limited applicability in the context of workflows due to their inability to model the rich requirements of today's organizational processes adequately.

In this paper, our observations are based on our experience in modeling and development efforts for a real-world workflow application for immunization tracking [SKM+96, PLS+96], experience in trying to use flexible transactions in multi-system telecommunication applications [ANRS92], and our understanding of the current state of workflow technology and its applications in real-world scenarios [SKM+96, MMC96, BSR96, ANRS92, VHL96].

Traditional database transactions provide properties such as failure atomicity, concurrency control and isolation. Most ATMs discuss controlled relaxation of these to gain more flexibility and better model real-world requirements. These are very useful concepts that could be applicable in workflows to a particular task (e.g., a transactional task [KS95] that interacts with a DBMS), a group of tasks supporting the two-phase commit protocol, or to the communication between tasks and the workflow scheduler [WWWD96]. Also, there is a potential need for concurrency control for workflows [JNRS93]. However, based on our review of requirements of existing applications using workflows (see section 2), we feel that transactional features form only a small part of a large-scale workflow application. Workflow requirements far exceed those of ATMs in terms of modeling, co-ordination and run-time requirements. It would definitely be useful to incorporate transactional semantics such as recovery and relaxed atomicity and isolation to ensure reliable workflow executions. Nevertheless, to view a workflow as an ATM, or to use ATMs to model workflows would be inappropriate. We do not think that ATMs provide a comprehensive or a sufficient framework for modeling large-scale enterprise-wide workflows, and hence they do not have too much to offer in terms of the application perspective.

This paper emphasizes the need for looking beyond the framework of ATMs for modeling and executing workflows. The notion of *transactions* as it exists in the database community is one of many similar terms that are part of the terminology of organizational processes today. For example, Electronic Data Interchange (EDI) transactions are used for defining interfaces and data formats for exchange of data between organizations; Health Level 7 (HL7) transactions are used to transfer patient data between health care organizations. Other uses are further discussed in section 3. Workflow systems should evolve with the needs of the business and scientific user communities, and provide the ability to incorporate additional functionality and semantics into the workflow model and run-time as and when required. Of course it is possible that in some specific domains, ATM based workflow models may be sufficient; however, we believe, such cases would be very few.

The organization of this paper is as follows. In section 2 we discuss four large-scale workflow applications that have been discussed in the literature to briefly review their major requirements. Section 3 provides a perspective into the characteristics and interpretation of *transactions* as they exist in workflow applications today. Section 4 provides a discussion of features offered by advanced transaction models and the primary requirements of workflow systems. Finally, we conclude the paper with our observations.

2 Review of Large Scale Workflow Applications

In this section we review some of the large-scale real-world applications of workflow technology that have been documented in the literature.

2.1 Workflow in Telecommunications

In [ANRS92], the authors describe the design and implementation of a service order provisioning workflow application prototype using *Flexible Transactions* [ELLR90, RELL90] in multi-database environments. It involves multiple, autonomous and heterogeneous databases and application systems that are required to service telephone customer requests. The service provisioning requires coordinated execution of Flexible Transactions at these information systems and exchange of data between them in a flexible and efficient manner.

A Flexible Transaction is specified using a set of subtransactions related by a set of execution dependencies among them. Each Flexible Transaction has a set of acceptable states defining the conditions for the success of the Flexible Transaction. This model relaxes the atomicity and isolation requirements of traditional transactions.

The major challenges of this project were:

1. providing a high level model so that workflows could be designed without too much modification of the application layer;
2. ensuring an interoperable system so that applications and DBMS could be easily added in the future; and
3. automation of the service provisioning process.

Flexible Transactions were found to be moderately useful in modeling the service provisioning workflow. According to the authors, one of the major obstacles in this project was the request for manual assistance due to errors, failures, and data inconsistencies. It was desired that the workflow be resilient to errors and failures and that the handling of these features be more automated. In hindsight, two observations can be made: (a) the exercise of modeling the server order processing workflow using the Flexible Transaction model was deemed moderately successful because the Flexible Transaction model was more flexible compared to other ATM alternatives. This perhaps puts less restriction as it relaxes both atomicity and isolation, thus supporting fewer of the transactional properties. However, (b) the ATMs including the Flexible Transaction model, lacked in several areas of meeting real-world requirements or needed flexibility. Key among the latter issues are: (a) the ability to model heterogeneous tasks that are not just subtransactions with the atomicity property, and especially the ability to model human participation in a workflow (which led to modeling of heterogeneous transactional and non-transactional tasks in the METEOR model [RS95b, KS95], (b) the ability to explicitly model a distributed system environment (which led to explicit specification of interface components to allow execution of a task in more than one way, and to model errors associated with execution of a task through a given interface [KS95]), (c) the ability to support error handling from specification to run-time support (for which there is still little work). In a subsequent study of telecommunication applications, requirements for support for dynamic workflow (including dynamic addition for tasks and dependencies) were found which lead to some initial steps in supporting dynamic workflows in [KS95].

2.2 Workflow in Managed Health care

The METEOR$_2$ multi-paradigm transactional WFMS is used for supporting state-wide immunization tracking requirements in the State of Connecticut [SKM+96, PLS+96]. This real-world application provides a rich set of requirements for supporting heterogeneous tasks (both human and application), a distributed infrastructure spanning autonomous enterprises, various communication paradigms (CORBA[OMG95a, OMG95b], Web and Lotus Notes [Cor96]), and multiple informational resources (DBMSs and Web resources). Organizations involved in this workflow application include health clinics, hospitals, insurance agencies, health care organizations and other agencies such as the State Department of Health (SDOH), Connecticut Healthcare Research and Education Foundation (CHREF), and Heath Maintenance Organizations (HMO).

This system can be described in terms of the following three components [SKM+96]:

- Databases: These include the central Master Patient Index (MPI), the central Master Encounter Index (MEI), the central Immunization (IMM) database, the central Eligibility (ELG) database, and the local Detailed Encounter (ENC) databases at provider locations. The MPI, MEI and IMM databases are located and managed by CHREF; the ELG database is maintained by the SDOH and provides access to information via EDI transactions.
- Clinical Workflow: This part of the workflow component primarily involves activities (or tasks) that are performed within a clinic or a hospital with respect to delivery of immunization vaccines to children. The workflow models the organizational structure of hospitals or clinics in terms of distinct roles for the admit clerk, triage nurse, physician assistant and doctor. Worklists are provided to facilitate and streamline human tasks. Medical alerts (e.g., delinquent immunizations) and contraindicators, and verification of treatment eligibility of patients are generated automatically by tasks within the workflow.
- Tracking Workflow: The tracking workflow serves to generate reports (e.g., list of children missing immunizations categorized by age, counties, etc.). These reports are made available on-line to physicians and health care administrators (at CHREF, SDOH and HMOs). More importantly immunization tracking involves reminding parents and case workers about children who are due or overdue for immunizations.

Workflow requirements, as outlined in [SKM+96] include:

1. support for heterogeneous computing environments (in terms of both hardware, software and communication infrastructures)
2. support for distributed client/service based architecture in which multiple organizations are involved, in terms of where the data is both stored and accessed;
3. support for inter- and intra-enterprise wide coordination of workflow tasks;
4. support for variety of tasks: transactional, non-transactional, user and application;
5. low cost to end users;

6. ease of modification, scalability, extensibility, fast design-to-implementation of the workflow system;
7. use of electronic data standards in the form of EDI transactions for informational exchange between autonomous organizations where possible; and
8. security, authentication and authorization support for patient-data communication and access.

The WFMS used CORBA for providing the infrastructure for distribution of tasks within an enterprise and for wrapping heterogeneous information resources (databases), and the Web for providing a uniform interface for human interaction with the WFMS. A version of the system that uses the Web instead of CORBA has also been developed for application developers who wish to avoid using one more technology and product for infrastructure support. However, the Web version has a set of fewer features and capabilities than can be more easily supported using CORBA compared to the Web. The application is implemented on a testbed distributed over machines at both CHREF and the University of Georgia's Large Scale Distributed Information Systems Lab (an on-line demonstration of a version is accessible at *http://lsdis.cs.uga.edu/demos*).

2.3 Workflow in a Business License Office

Workflow and imaging technology is being used in the Clark County Department of Business License for automating and streamlining the licensing system and to turn the department into a "paper-less environment" [MMC96]. ActionWorkflow [Tec95] provided the workflow management infrastructure and Image X provided the imaging tools for this project.

Workflow requirements stemmed from the ineffectiveness of the previous system (which was primarily human-centered, paper based, and included an IBM ES 9000 mainframe program for information retrieval) to deal with the high demand for license renewals (approximately 72000 license renewals and applications annually). There was a need to automate the department and re-engineer business processes with emphasis on customer service. Some of the major requirements for their workflow solution included:

1. making the entire system more scalable to handle the high rate of customer influx;
2. elimination of paper to the maximum extent possible;
3. accountability of employees, and tracking the workflow process;
4. establishment of a horizontal flow of information and elimination of lost files;
5. better means of coordination between departments, as compared to mail and telephone;
6. reducing error rate of licensing data generated; and
7. ability to provide integration with the existing mainframe environment.

The image/workflow solution was effective in ironing many of the problematic workflow requirements, including providing a paper-less environment, and provided an automated, efficient license handling mechanism that was designed around the customer.

Technical challenges faced by the workflow integrators involved integration of various software, hardware and legacy application systems. More importantly, the major difficulties that were encountered during this project were not related to workflow aspects such as modeling and scheduling, or advanced transactional features such as correctness or recovery; the primary source of problems was the human and the bureaucratic structure that was in conflict to the re-engineering in the business process.

2.4 Workflow in a Genome Laboratory

The use of workflows in high-throughput mission-critical application systems such as tracking of experimental data at the Center for Genome Research is discussed in [BSR96]. This study is based on LabFlow-1 [BSR96], a database benchmark for high-throughput production WFMS. Workflow management is used in the laboratory information systems setting to automate the handling of samples, testing, instrumentation, data capture, and tracking of event histories. The DBMS is primarily used in this project to control and track sample collection workflows.

The throughput of such experiments range in the order of approximately 15,000 transactions per day, with peak rates reaching 22.5 queries and updates per second [BSR96]. To be effective in such high-throughput production environments, a lot is desired in terms of scalability, efficiency, and reliability of the underlying WFMS infrastructure. The authors address the requirements of the DBMS that forms one of the critical components of the WFMS. Some of the important requirements for the DBMS mentioned in [BSR96] are:

1. standard database features such as providing isolation, consistency, failure recovery, high-level query language and query-optimization;
2. support for maintaining audit histories of the workflow activity (workflow tracking);
3. ability to store complex-structured data; and
4. ability to allow dynamic modification of the schema at run-time as the workflow itself is characterized by dynamism in terms of modification of flow of control and modification of tasks.

Careful study of the requirements in the applications described above reveals that these are guided more by the organizational and human aspects of the operational environment rather than solely on transactional characteristics such as isolation, atomicity and consistency. Transactional semantics are desirable in many real-world applications (e.g., databases form a critical enabling component of the WFMS in the genome laboratory application described above); but do not form the framework for modeling most applications.

3 Perspective of an Academic Practitioner

As practicing researchers, the idea of using related transaction models was appealing to us. We felt that such a model could provide a rigor or structure that was lacking in the work on workflow management [ANRS92, BDSS93]. There are few, if any, examples of successes in developing systems that implement ATMs for significant commercial applications. The principles and theory associated with basic transaction models do not seem to scale very well with HAD environments that are typical to current workflow systems. However, the concept of transactions remains very attractive.

The notion of *transactions* and their ACID properties in the context of database systems is well known. Transactions emphasize data consistency and are limited to databases, and hence are too data-centric [BDSS93]. This limits their applicability and modeling power in capturing important aspects of non-traditional, large-scale, and multi-system applications. Requirements of such applications (as seen in section 2) that are beyond the purview of ATMs include: 1) capability to explicitly define the functionality and organizational structure of organizational process involved, 2) support of coordination and execution of tasks in heterogeneous intra- and inter-enterprise environments, 3) modeling and support for human involvement with the run-time system, and 4) error handling and failure recovery.

Workflow management is specifically defined to address these real-world challenges. It provides the tools to integrate humans, computer systems, information resources and organizational processes into a unified solution. Typical workflow solutions involve such interactions between application and human tasks. Hence, the requirements of workflow management systems are far more challenging than those faced by current database systems [AS96]. In workflow applications, database resources might comprise only a part of the entire solution. For a task that entirely interacts with a DBMS, executing it as a transaction is often a desirable choice. At the same time, workflows involve other user and application tasks (e.g., tasks that interact with legacy systems) that are non-transactional in nature.

Due to the wide acceptance and applicability of workflows to application domains that extend beyond transaction based (primarily database related) environments, the term *transaction* is being used in a more loose manner with various connotations. These interpretations are based on: 1) the type of tasks and processing entities that are part of the workflow process, 2) the application domain or semantics of the organizational process that is being modeled, 3) the communication infrastructure that is used to develop the WFMS, and 4) transactional or advanced transactional semantics (such as relaxed isolation and atomicity) that can be attributed to the tasks, sub-workflow, or the workflow as a whole. It is important to understand each of these interpretations to be able to appreciate the similarities and differences between transactions from the world of database systems and those involved in the realm of multi-system workflow management systems. Let us consider some of the frequently encountered interpretations for *transactions* in the context of WFMS:

1. **Task specific interpretation in databases and distributed transaction processing.** In general, a workflow task is considered to be a *black box* that is functional in nature, i.e., the functionality of the task is orthogonal to that of the workflow process [AKA+94]. The tasks themselves could be transactional or non-transactional in nature [RS95b, KS95]. Transactional tasks are those that minimally support the atomicity property and maximally support all ACID properties of traditional transaction models [MSKW96, KS95]. These

tasks typically include those that interact with a DBMS by using *BEGIN_TRANSACTION - END_TRANSACTION* semantics, contracts (stored procedures), and two-phase commit (2PC) tasks [Wan95, MSKW96] for synchronizing transactions across multi-DBMSs. In addition, tasks that use the *XA-Protocol* [GR93] based RPC to communicate with transactional processing entities such as a TP-monitor in a distributed environment [WWWD96] can also be included in this category. Non-transactional tasks are used to include applications that cannot ensure isolation or atomicity as a part of the workflow process. Such task types are commonplace in the real-world and involve activities requiring interaction with humans, legacy systems, and others that interface with other processing entities that do not provide transactional semantics support (e.g., HTTP servers, Lotus Notes, file systems, word processors, spreadsheets and decision support systems).

2. **Domain specific interpretation.** The move from a paper-based society to a paper-less one, and the increasing popularity of electronic commerce have led to evolution of standards for electronic data exchange across organizations. Some of these include *Electronic Data Interchange (EDI)* standards such as ANSI Accredited Standards Committee (ASC) X12 that are used in numerous commercial settings (e.g., ANSI 270 and 271 transactions for health care eligibility inquiry and response used in [SKM+96]), and the ANSI HL7 standard that is used specifically in the medical domain. The term *transaction* in this setting refers to the exchange of sufficient data in a standard electronic format necessary to complete a particular business action often using domain specific information. This view of a transaction tends to focus more on business requirements and contracts rather than on the need for maintaining data consistency within a database or to support atomicity or other transactional property between communicating processes or for a RPC call. Workflow technology is being applied in various forms to application domains such as manufacturing, banking, health care, and finance that use domain specific *transaction* formats extensively. One of the tasks within a workflow process could involve sending data from one information system to another using an EDI *transaction*. At the receiving end, another workflow task could write the data that it receives to a DBMS in a *transactional* (having ACID properties) manner. The semantics associated with each of these transactions are different. Hence, the WFMS would have to be designed so that it can deal with different *transaction* forms in an appropriate manner.

3. **Business-process specific interpretation.** Database transactions and transaction processing aim at preserving data consistency and ensuring reliability in case of faults and failures. These semantics cannot be applied directly to workflow systems since tasks within a workflow process are both transactional and non-transactional in nature. However, at the same time, workflow systems should be correct and reliable. Correctness and reliability in the case of workflow systems is more applicable from a broader perspective - that of the *organizational process* involved in addition to the data that forms a part of the process. According to [EL95], a workflow *transaction* should ensure consistency from the business process point

of view. This notion of a transaction is broader and less formal from systems perspective than that of traditional transactions since it proposes to support the business process flows in a consistent manner as desired by the workflow designer. Implementation support for such a concept would require an additional layer of control than that provided in transaction processing since workflows include features (e.g., roles, worklists, error handling) that are not available in (advanced) transaction models and transaction processing systems.

4. **Infrastructure specific interpretation.** Workflow management systems are large-scale applications that can be implemented using various infrastructure technologies such as Customized Transaction Management (CTM) [GHS95], Distributed Object Management specifically using CORBA [MHG+92, GHKM94, MSKW96, SKM+96, WWWD96, SJKB94], World Wide Web [PLS+96, SKM+96, Tec95], TP-monitors [WWWD96], Lotus Notes [RM96] and security services (as in *secure transactions* supported in the electronic commerce and Web-based services). The concept of *transactions* has been addressed in many of these technologies to some extent. For example CORBA provides an Object Transaction Service as a part of the Common Object Services Specification [OMG95b] that enables objects in distributed environments to take part in a *transactional context*; TP-monitors also provide transactional semantics in a distributed environment. The HTTP protocol used in the Web paradigm, on the other hand, does not provide any transactional semantics. Hence, we see that different interpretations of transactions are supported by each of these infrastructures.

From the above discussion, it is important to observe that the notion of transactions in workflow management is more generalized as compared to that in transaction processing systems and its interpretation could involve various variables associated with factors mentioned above. *Unlike advanced transaction systems, WFMS interact with database systems if required as part of the organizational process, however, this is not their primary focus.*

4 Discussion

An early ATM model proposed was the Saga model [GMS87, GMSG+91]. The limited applicability of the Saga model for modeling workflow systems is due to the fact that Sagas provide control flow and compensation that might be too limited to represent the exact semantics of the workflow constraints. Other ATMs (e.g., Flexible Transactions [RELL90, ELLR90], ConTracts [WR91, RS95a], ATM [DHL91]) have been proposed to overcome the limitations of the Saga model by supporting mechanisms to specify more general control flow, recovery policies and to provide support for long running activities. Often ATMs are developed with specific applications in mind, and are based on fixed control and data dependencies, using atomic transactions as the building blocks, and are therefore limited in their applicability in dynamic workflow applications of today. Moreover, ATMs are executed in relatively uniform environments and primarily deal with automated tasks. Also, workflow features such as task-level interdependencies, interface-level details, human involvement, integration of transactional and non-transactional tasks, organizational roles, error-handling, and support for dynamism, interoperability and mobile clients find very limited support, if any, in ATMs (see the appendix for a normative comparison of ATMs and workflow systems).

Distributed transaction processing and on-line transaction processing focus on the efficient execution of relatively simple tasks without any support for coordination between heterogeneous tasks. They are still far from the degree of flexibility required by long-running organizational processes that operate in a heterogeneous environment. TP-monitors [GR93] in a multi-database environment provide a limited form of coordination. They are inadequate to address the issues involved in dealing with multiple tasks (both user and application) that are executed on distributed processing entities since they require that all the resources in the distributed environment should support a two-phase commit (2PC) protocol - a requirement which is not satisfied by processing entities such as file systems, legacy applications, and humans.

In our perspective, the role of ATMs in workflow systems is of a supportive nature. Advanced transaction modeling concepts are quite restricted in terms of being directly applicable in process-oriented, large-scale workflow applications that run in HAD environments. Workflow systems today are still weak in terms of characteristics such as fault-tolerance, consistency, and in their support for recovery in case of exceptions and failures. ATMs have solved most of these problems in the domain of database systems. Research in the areas of workflow systems can benefit from these approaches from a conceptual point of view.

Transactional semantics such as atomicity and isolation in their strict sense are not practical in workflow systems since tasks in a workflow domain are generally long-lived and could themselves be non-transactional in nature. Many of the solutions for recovery in transaction processing systems can be used to address recovery issues in workflow systems: for example, advanced transaction concepts such as compensating transactions can be mapped to the workflow domain in terms of a compensating task that could be used to *undo* (often partially) what was done by an incomplete task during system failure; logs similar to those in transaction processing could be maintained for recording the history of the workflow process, thereby aiding in the recovery process [KS95, AKA+94, EL96].

To address many of these advanced issues, workflow systems should borrow ideas that have been used effectively in concurrent, large-scale distributed and database systems, but, they should not rely entirely on them as many of these systems have developed models for environments that are limited in scope as compared to that in workflow systems.

5 Conclusion

In conclusion, we summarize the observations we have made in this paper:

- There are several interpretations for *transactions* in organizational processes today and all of them may need to be accommodated in a workflow technology that supports these processes.

- Features offered by ATMs meet a very restricted subset of requirements of large-scale enterprise-wide workflow systems.
- We do not see ATMs as being a primary basis for modeling and executing workflow systems that have real-world commercial applicability. However these models provide useful features (e.g., relaxed atomicity, relaxed isolation, concurrency control and recovery) which can be used in components (e.g., tasks) that form a part of a WFMS.

There is a need for multi-disciplinary research to address the challenging issues raised by emerging workflow technology. Humans are an essential part of any organizational process, and human work involves many diverse issues. Therefore, research involving expertise from multiple disciplines is most likely to bring the highest return. Information is another critical asset of any organization; we believe that more human-centric approaches with integral support for information management are needed for a successful workflow technology. We should, therefore, look beyond the capabilities provided by transaction processing systems and ATMs in modeling the complexities of large-scale, mission-critical workflow applications of the future.

References

[AAA+96] G. Alonso, D. Agarwal, A. E. Abbadi, M. Kamath, and R. Gunthor. Advanced transaction models in workflow contexts. In *Proc. of 12th. IEEE Intl. Conference on Data Engineering*, New Orleans, LA, February 1996.

[AKA+94] G. Alonso, M. Kamath, D. Agarwal, A. El Abbadi, R. Gunthor, and C. Mohan. Failure handling in large scale workflow management systems. Technical Report RJ9913, IBM Almaden Research Center, November 1994.

[ANRS92] M. Ansari, L. Ness, M. Rusinkiewicz, and A. Sheth. Using flexible transactions to support multi-system telecommunication applications. In *Proc. of the 18th Intl. Conference on Very Large Data Bases*, pages 65–76, August 1992.

[AS96] G. Alonso and H.J. Schek. Research issues in large workflow management systems. In *[SGJ+96]*, Athens, GA, May 1996.

[BDG+94] A. Biliris, S. Dar, N. Gehani, H. Jagadish, and Krithi Ramamritham. Asset: A system for supporting extended transactions. In *Proc. of ACM SIGMOD Conference on Management of Data*, pages 44–54, Minneapolis, MN, May 1994.

[BDSS93] Y. Breitbart, A. Deacon, H. Schek, and A. Sheth. Merging application-centric and data-centric approaches to support transaction-oriented multi-system workflows. *SIGMOD Record*, 22(3):23–30, September 1993.

[BSR96] A. Bonner, A. Shruf, and S. Rozen. Database requirements for workflow managment in a high-thoroughput genome laboratory. In *[SGJ+96]*, Athens, GA, May 1996.

[CD96] Q. Chen and U. Dayal. A transactional nested process management system. In *Proc. of 12th. IEEE Intl. Conference on Data Engineering*, New Orleans, LA, February 1996.

[Coa94] The Workflow Management Coalition. Glossary – a workflow management coalition specification. Technical report, The Workflow Management Coalition, Brussels, Belgium, November 1994.

[Cor96] Lotus Development Corporation. Lotus Notes Products. Technical report, Lotus Development Corporation, March 1996. URL: http://www.lotus.com/home/notes.htm.

[CR90] P. Chrysanthis and K. Ramamritham. ACTA: A framework for specifying and reasoning about transaction structure and behavior. In *Proc. of ACM SIGMOD Conference on Management of Data*, pages 194–203, Atlantic City, NJ, 1990.

[CR92] P. Chrysanthis and K. Ramamritham. *ACTA: The SAGA Continues*, chapter 10. Morgan Kaufman, San Mateo, CA, 1992.

[DHL91] U. Dayal, M. Hsu, and R. Ladin. A transactional model for long-running activities. In *Proc. of the 17th Intl. Conference on Very Large Data Bases*, pages 113–122, Barcelona, Spain, September 1991.

[EL95] J. Eder and W. Liebhart. The workflow activity model WAMO. In *Proc. of the 3rd Int. Conf. on Cooperative Information Systems*, Vienna, Austria, May 1995.

[EL96] J. Eder and W. Liebhart. Workflow recovery. In *Proc. of the First IFCIS Conf. on Cooperative Information Systems*, Brussels, Belgium, June 1996.

[ELLR90] A. K. Elmagarmid, Y. Leu, W. Litwin, and M. Rusinkiewicz. A multidatabase transaction model for interbase. In *Proc. of the 16th Intl. Conference on Very Large Data Bases*, pages 507–518, August 1990.

[Elm92] A. Elmagarmid. *Database Transaction Models for Advanced Applications.* Morgan Kaufmann Publishers, CA, 1992.

[GH94] D. Georgakopoulos and M.F. Hornick. A framework for enforceable specification of extended transaction models and transactional workflows. *International Journal of Intelligent and Cooperative Information Systems*, 3(3):599–617, 1994.

[GHKM94] D. Georgakopoulos, M. Hornick, P. Krychniak, and F. Manola. Specification and management of extended transactions in a programmable transaction environment. In *Proc. of the Tenth International Conference on Data Engineering*, Houston, TX, February 1994.

[GHS95] D. Georgakopoulos, M. Hornick, and A. Sheth. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2):119–154, April 1995.

[GMS87] H. Garcia-Molina and K. Salem. Sagas. In *Proc. of ACM SIGMOD Conference on Management of Data*, pages 249–259, San Francisco, CA, May 1987.

[GMSG+91] H. Garcia-Molina, K. Salem, D. Gawlick, J. Klein, and K. Kleissner. Modeling long-running activities as nested sagas. *IEEE Data Engineering Bulletin*, 14(1), March 1991.

[GR93] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques.* Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[JAD+94] S. Joosten, G. Aussems, M. Duitshof, R. Huffmeijer, and E. Mulder. *WA-12: An Empirical Study about the Practice of Workflow Management.* University of Twente, enschede, The Netherlands, July 1994. Research Monograph.

[JNRS93] W. Jin, L. Ness, M. Rusinkiewicz, and A. Sheth. Concurrency control and recovery of multidatabase work flows in telecommunication applications. In *Proc. of ACM SIGMOD Conference*, May 1993.

[KR96] M. Kamath and K. Ramamritham. Bridging the gap between transaction management and workflow management. In *[SGJ+96]*, Athens, GA, May 1996.

[KS95] N. Krishnakumar and A. Sheth. Managing heterogeneous multi-system tasks to support enterprise-wide operations. *Distributed and Parallel Databases*, 3(2):155–186, April 1995.

[MAGK95] C. Mohan, G. Alonso, R. Guenthoer, and M. Kamath. Exotica: A research perspective on workflow management systems. *Data Engineering Bulletin, Special Issue on Infrastructure for Business Process Management*, 18(1), March 1995.

[MHG+92] F. Manola, S. Heiler, D. Georgakopoulos, M. Hornick, and M. Brodie. Distributed object management. *Intnl. Journal of Intelligent and Cooperative Information Systems*, 1(1), 1992.

[MMC96] R Medina-Mora and K. W. Cartron. ActionWorkflow in Use: Clark County Department of Business License. In *Proc. of the Twelfth International Conference on Data Engineering*, New Orleans, LA, February 1996.

[MSKW96] J. A. Miller, A. P. Sheth, K. J. Kochut, and X. Wang. Corba-based run time architectures for workflow management systems. *Journal of Database Management, Special Issue on Multidatases*, 7(1), 1996.

[OMG95a] OMG. The common object request broker: Architecture and specification, revision 2.0. Technical report, Object Management Group, July 1995.

[OMG95b] OMG. CORBAservices: Common object services specification. Technical report, Object Management Group, March 1995.

[PLS+96] D. Palaniswami, J. Lynch, I. Shevchenko, A. Mattie, and L. Reed-Fourquet. Web-based multi-paradigm workflow automation for effiecient health-care delivery. In *[SGJ+96]*, Athens, GA, May 1996.

[PPVW96] D. Perry, A. Porter, L. Votta, and M. Wade. Evaluating workflow and process automation in wide-area software development. In *[SGJ+96]*, Athens, GA, May 1996.

[RELL90] M. Rusinkiewicz, A. Elmagarmid, Y. Leu, and W. Litwin. Extending the transaction model to capture more meaning. *SIGMOD Record*, 19(1):3–7, 1990.

[RM96] B. Reinwald and C. Mohan. Structured workflow management with lotus notes release 4. In *Proc. of 41th IEEE Computer Society International Conference*, pages 451–457, Santa Clara, CA, February 1996.

[RS95a] A. Reuter and F. Schwenkreis. Contracts - a low-level mechanism for building general-purpose workflow management systems. *IEEE Data Engineering Bulletin*, 18(1), 1995.

[RS95b] M. Rusinkiewicz and A. Sheth. Specification and execution of transactional workflows. In W. Kim, editor, *Modern Database Systems: The Object Model, Interoperability and Beyond*, pages 592–620. ACM Press, New York, NY, 1995.

[SGJ+96] A. Sheth, D. Georgakopoulos, S. Joosten, M. Rusinkiwicz, W. Scacchi, J. Wileden, and A. Wolf. A multidisciplinary research agenda for work activities coordination: Report from the nsf workshop on workflow and process automation in information systems. Technical report, To be published, July 1996. URL: http://lsdis.cs.uga.edu/activities/NSF-workflow.

[She95] A. Sheth. Tutorial notes on workflow automation: Application, technology and research. Technical report, University of Georgia, May 1995. presented at ACM SIGMOD, San Jose, CA, URL: http://lsdis.cs.uga.edu/publications.

[SJKB94] H. Schuster, S. Jablonski, T. Kirsche, and C. Bussler. A client/server architecture for distributed workflow management systems. In *Proc. of the Third International Conf. on Parallel and Distributed Information Systems*, pages 253–256, Austin, TX, September 1994.

[SKM+96] A. Sheth, K. J. Kochut, J. Miller, D. Worah, S. Das, C. Lin, D. Palaniswami, J. Lynch, and I. Shevchenko. Supporting state-wide immunization tracking using multi-paradigm workflow technology. In *Proc. of the 22nd. International Conference on Very Large Databases*, Bombay, India, September 1996.

[Smi93] T. Smith. The future of workflow software. *INFORM*, pages 50–51, April 1993.

[Tec95] Action Technologies. Metro tour. Technical report, Action Technologies, Inc., 1995. URL: http://www.actiontech.com/.

[TV95] J. Tang and J. Veijalainen. Transaction-oriented work-flow concepts in inter-organizational environments. In *Proceedings of the 4th. Intnl. Conference on Information and Knowledge Management*, Baltimore, MD, November 1995.

[VHL96] B. Vivier, I. Haimowitz, and J. Luciano. Workflow requirements for electronic commerce in a distributed health care enterprise. In *[SGJ+96]*, Athens, GA, May 1996.

[Wan95] X. Wang. Implementation and performance evaluation of CORBA-based centralized workflow schedulers. Master's thesis, University of Georgia, August 1995.

[Wei93] G. Weikum. Extending transaction management to capture more consistency with better performance. Technical report, Department of Computer Science, ETH, Zurich., 1993.

[WF94] T. White and L. Fischer. *The Workflow Paradigm - The Impact of Information Technology on Business Process Reengineering.* Future Strategies, Inc., Alameda, CA, 1994.

[WR91] H. Wachter and A. Reuter. *The ConTract Model*, chapter 7. Morgan Kaufmann, 1991.

[WWWD96] D. Wodtke, J. Weissenfels, G. Weikum, and A. K. Dittrich. The mentor project: Steps towards enterprise-wide workflow management. In *Proc. of 12th. IEEE Intl. Conference on Data Engineering*, New Orleans, LA, February 1996.

Appendix: A Normative Perspective

	Advanced Transaction Models	**Workflow Systems**
Theoretical Foundation	Usually good theoretical basis.	Weak dependency, except for scheduling components. Driven by practical considerations.
Granularity	Transactions.	Tasks, activities, or steps
Methodology	Data-centric. Emphasis on data consistency.	Process-centric. Emphasis on task dependency.
Correctness Criteria	Serializability, if at all possible.	Primitive, often limited to scheduling.
Failure Atomicity	Inherent.	Not part of most models.
Concurrency Control	Inherent.	Limited support.
Recovery	Well-defined. *Rollback* and *compensation.*	Insufficient support. *Forward recovery* when supported.
Error Handling	Limited.	Limited.
Task/Activities	Supports transactions only.	Supports both human and application tasks.
Processing Entities	Usually DBMS.	Heterogeneous systems (e.g., DBMS, legacy applications, GUI)
Coordination Support	Limited.	Inherent.
Modeling of Organizational Structure	Usually absent.	Usually inherent.
Worklists	No support.	Strong support.
Flexibility	Varied.	Good.
Implementation Status	Very few exist.	Numerous commercial products and few research prototypes.
Applicability to Non-DBMS applications	Limited.	Extensive.