# AN ARCHITECTURE FRAMEWORK FOR DISTRIBUTED WORKFLOW MANAGEMENT

*Adel A. El-Zoghabi*

Department of Information Technology, Institute of Graduate Studies and Research
Alexandria University, P.O. Box 832 El-Shatby, Alexandria, Egypt
E-mail: zoghabi@igsrnet.net

**ABSTRACT**

The main purpose of this paper is to present an architecture framework for business process reengineering based on distributed workflow management. The proposed framework is primarily based on a multi-tiered architecture, which are the front-end desktop client machines, the middle tier application servers, the back-end servers, and the configuration server. This architecture delivers the benefits of the client/server computing paradigm and the Web technology in a distributed manner. The application logic and processing are embedded on the middle tier application servers instead of on the desktop client machines. The check verification business process is adopted as a case study in order to illustrate the operation and the benefits of the proposed multi-tiered architecture.

***Keywords***: Workflow management systems (WFMS), Distributed information systems, Business process reengineering, Office automation.

## INTRODUCTION

The advent and widespread exposure of a truly global economics has immersed all businesses into an intensively competitive environment moving with accelerating rates of changes. Gradual improvements in productivity and enhancement in quality are no longer enough to maintain market leadership. The fast delivery of new products/services and the rapid modification of existing applications are key survival factors. These requirements have forced enterprises to constantly reconsider and optimize the way of doing business [1,2].

Early computer applications have greatly increased productivity and have provided better services. Despite their great success, the early versions of computer applications have two major bottlenecks [2]. First, All business policies and information accesses were coded into the application program. These systems were hard to maintain and to enhance especially when the business policies and data were changed. The database technology has partially solved this problem by separating data accesses from the applications. Nevertheless, business policies are still coded and any changes would require modifying the application code. Second, the traditional computer applications were designed and developed to work independently to solve specific problems. The currently available network and distributed computing technologies have made it possible for them to collaborate and exchange messages. A new robust technology for integrating these processes islands at a higher level so that they can collaborate to provide business solutions is urgently required.

Workflow technology has been proposed to address the above problems of early computer applications. The realization of this new technology is reached by providing methodologies and software to support: (1) business process modeling to capture workflow as workflow specifications, (2) business process reengineering to optimize specified processes, and (3) workflow automation to generate workflow implementations from workflow specifications [2,3]. The key concept behind workflow technology is the need for business process reengineering whose purpose is to

increase productivity, reduce cost, and respond to the changing environment more quickly. The development of new technologies such as distributed computing, object technology, and distributed database that facilitate open and reliable information exchange and collaboration across the organization is another driving force for the promotion of workflow technology [4,5,6].

The workflow concept has evolved from the notion of process in manufacturing and office. The search to increase efficiency by concentrating on the routine aspects of work activities makes it possible to separate the work activities into well defined tasks, roles, and procedures which regulate most of the work in organization or office. With the introduction of Information Technology, processes in work place are partially or totally automated by information systems. Database, transaction processing, and distributed systems technologies provide the basic infrastructure for supporting information processes [7].

The main purpose of this paper is to present an architecture framework for business process reengineering based on distributed workflow management. The proposed architecture is intended to be more reliable in the presence of concurrent activities or failures. This paper is organized as follows: the next section illustrates the basic structure of most workflow management systems placing emphasis on both the traditional models for workflow management systems and their most recently adopted reference models. Then, we focus on transactional processing and concurrency control issues in workflow management systems and how they differ from traditional database systems, followed by the presentation of the proposed multi-tiered distributed workflow management system. Also, an overview is given for the kind of business processes that contain both manual and automated activities in which the check verification workflow system is considered for automation. Finally, the conclusions of the work are presented.

## WORKFLOW MANAGEMENT SYSTEMS

A workflow process is a coordinated set of activities that are connected in order to achieve a common business goal [2]. An activity is defined as a logical step or description of a piece of work that contributes toward the accomplishment of a process. A single activity may be conducted either manually or automatically. A workflow process is first specified using a process definition language and then executed by a Workflow Management System (WFMS) [8,9]. A WFMS is a system that completely defines, manages, and executes workflow process through the execution of software whose order of execution is defined by a computer representation of the workflow logic [3]. Figure 1 illustrates the traditional model of workflow management systems.

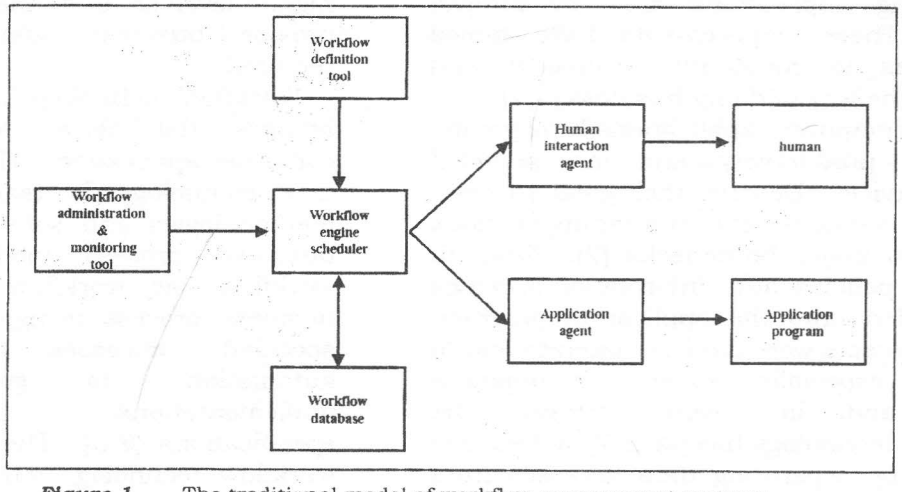


***Figure 1*** The traditional model of workflow management systems

Many commercial systems have been introduced to support workflow management by carrying out document management, imaging, application launching, human coordination, collaboration, and co-decision [3,9,10]. However, the commercial systems suffer from the following disadvantages: (1) limited inter-operability, (2) may not ensure correctness and reliability of applications in the presence of concurrency and failures, and (3) suffer from performance and scalability problems [11,12,13].

To efficiently support workflow management, organizations must evolve their existing computing environments to a new distributed environment. Distributed object management must be investigated and utilized to allow workflow management systems to cope with replacement migration and evolution of heterogeneous, autonomous, and distributed systems [8].

The lack of standards has been one of the major obstacles to the commercialization of workflow technologies. Workflow developers have their own workflow model, specification language, and Application Program Interface (API). The past few years have witnessed a significant progress in the establishment of workflow-related standards [5]. Examples of these standards are WFMC, MAPI-WF, and ODBC. Enabling distributed technologies such as e-mail, CORBA, and ActiveX/DCOM are also among the standard technologies affecting the workflow research and products [8].

WFMC that stands for the Workflow Management Coalition was founded in 1993 and since that is considered as the standard body for all workflow systems. The standard work of WFMC is focused around the workflow reference model illustrated in Figure 2. The reference model specifies a framework for the workflow systems with all its characteristics, functions, and interfaces. The main issue is to specify the five APIs that surround the workflow engine. These APIs provide a standard means for communication between the workflow engines and clients.

MAPI-WF which stands for message API Workflow Framework, is Microsoft's initiative to the WFMC. The idea is to combine the functionality of workflow systems and the flexibility of messaging systems so that applications that span both messaging users and applications can be deployed. In a messaging environment, a workflow request can be packed within some body part of a message. MAPI-WF provides a standard set of body parts and properties so that workflow packages can be delivered to and from workflow engine. Workflow components such as workflow engine, workflow applications, and workflow tools that conform to the WFMC-WF can communicate via various types of messaging systems [5].

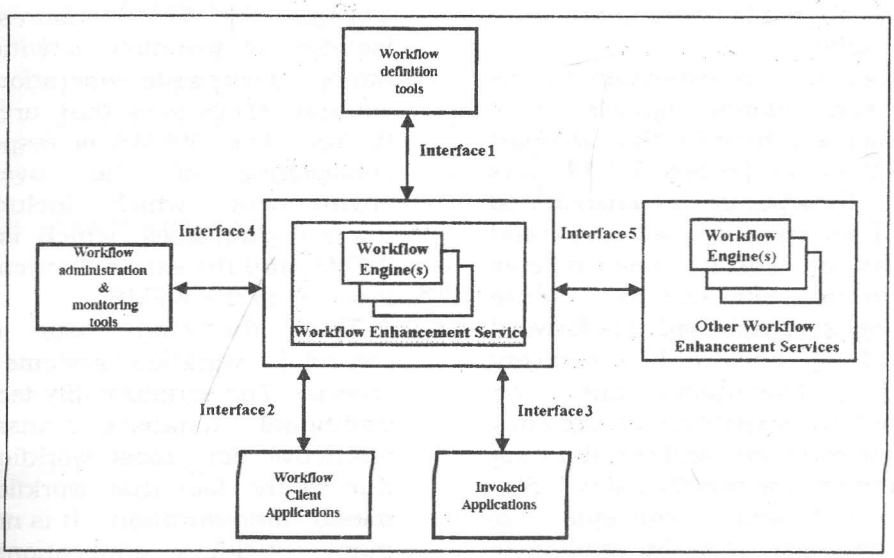


***Figure 2*** The wfmc workflow reference model

Recently, two new technologies for workflow systems have been advised, these are the object technology and the distributed computing technology. In contrary with traditional database management systems, workflow management systems are distributed and extendable by nature. In order to accomplish a workflow task, the workflow engine needs to invoke remote applications. Object and Distributed computing technologies such as CORBA, which stands for Common Object Request Broker Architecture, and ActiveX/DCOM, which stands for Distributed Components Object Module are vital in executing remote and heterogeneous applications.

## TRANSACTIONS AND CONCURRENCY CONTROL IN WFMS

Transaction processing, as first introduced for database applications, is an execution processing with ACID properties [6]. The keyword ACID came from the four properties that govern the execution of a transaction. First, either all or none of the transaction's effects take place (Atomicity). Second, the transaction must map the database from one consistent state to another consistent state (Consistence). Third, multiple transactions may be executed concurrently but the overall effect must be equivalent to some sequential execution (Isolation). Fourth, the effects of executing the transaction are made permanently once committed (Durability).

Many researchers, as extension to the conventional transactional models, have investigated workflow models that support certain transactional properties [1,7,14]. It is possible to incorporate transactional semantics such as recovery, atomicity and isolation to ensure correct and reliable workflow execution. For example failure atomicity can be accomplished via forward recovery and backward recovery compensation [11]. Execution atomicity can also be ensured by specifying consistency units of workflow processes and coordinating their execution to ensure serializability [15].

Although traditional concepts of transactional database can be applied to workflow processes, several bottlenecks still surrounding the research efforts. First, a workflow activity is more complicated than a database transaction and may involve human interactions, heterogeneous data, and distributed components. Second, the structure of a workflow process is more complicated than a database transaction and the execution of a process may establish quite complex control and data flow dependencies among activities of the process. Third, a workflow process specification may include concurrent execution of activities [16].

The primary purpose of concurrency control in traditional database is to ensure execution isolation of a transaction from other conflicting transactions. Many research efforts have been conducted concerning concurrency control in WFMS [7,17]. Although the outcome of most research illustrated that concurrency control is either unnecessary or too costly for many workflow applications, concurrency control is very important for some workflow applications where critical operations require a consistent view of the execution environment [18].

The main focus of concurrency control in workflow systems is a little bit different from that of database systems. In database systems, the main objective is to guarantee the execution isolation of transactions, which include atomic read/write operations that are visible to the DBMS. In workflow systems, the WFMS ensures the execution isolation of workflow activities that include atomic read/write operations as well as external executions that are invisible to the WFMS. The WFMS is responsible for the consistency of the overall execution environment, which includes both the internal databases, which is visible to the WFMS and the external execution, which are invisible to the WFMS.

Thus, the main issue of concurrency control in workflow systems is correctness criteria. The serializability technique used in traditional database transactions is too restrictive for most workflow applications due to the fact that workflow activities are mostly long-duration. It is not acceptable in many workflow applications to schedule conflicting activities sequentially as for

read/write operations in database transactions. Some existing research addresses the problem by specifying and enforcing data and execution dependencies among workflow activities [6,16,17]. Check-in & check-out schemes are suitable for workflow in the engineering environment such as CAD/CAM and CASE, where decisions to access objects are more adhoc.

Some other research efforts use database techniques but allow flexible specification of consistency requirements with respect to scope and granularity. An example is given in [16] which allows grouping a collection of workflow activities of a workflow process into a consistency unit and uses traditional concurrency control to ensure isolation of consistency units in terms of serializability. Enforcing proper data and execution dependencies ensures correct execution of activities inside a consistency unit.

## PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture for Distributed Workflow Management Systems over a TCP/IP network is based on a multi-tiered-computing model. The diagram of Figure 3 illustrates this model.

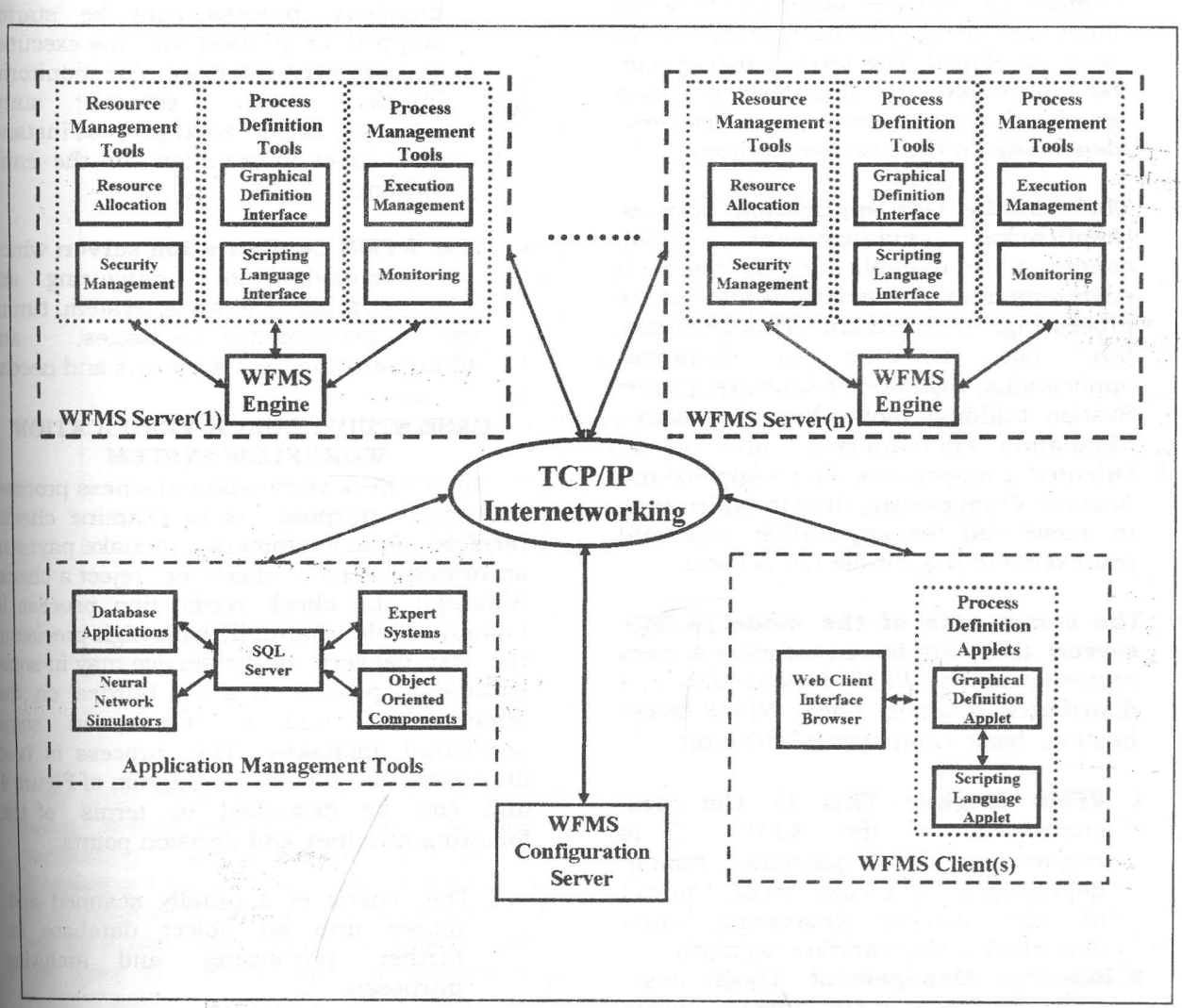


***Figure 3*** The proposed multi-tiered distributed wfms

According to the above diagram, the main components that constitute the model functionality are:

- **The client side of the model (WFMS client):** is either a standard Java-Enabled Web browser with process definition browsing facilities or a specialized process definition tools. The standard process definition tools include both a graphical definition component and a scripting language interface. Two Java applets, one for the graphical definition component and the other for the script language definition component, have been developed. The script language can be interpreted and transformed into a graphical representation and vice versa depending on the user preferences.

- **The middle tier application servers (Application management tools):** contain a set of application development environments that carries out most of the processing requirement. This includes, but not restricted to, Database applications, SQL Server software, Expert System building tools, Neural Networks Simulation environments, and Object-Oriented Components (Java Applets and Active X Components). It is intended here to move all the application logic and processing to the middle tier servers.

- **The server side of the model (WFMS server)** this can be one of more servers representing the WFMS functionality in a distributed manner. Each WFMS server has four basic components which are:

  i. **WFMS Engine: This is the core** component of the WFMS. It is responsible for sequencing, timing, dependency, and concurrency control of the various processing units described in the workflow scenario.

  ii. **Resource Management Tools:** these tools are used at run time to allocate execution resources to a task based on rules set by the organization. Authorization and Authentication of users are also performed by the security management interface.

  iii. **Process Definition Tools:** the business process is defined via the graphical definition applet or the scripting language applet from within the client Java-Enabled Web browser. This definition is then submitted to either the graphical definition interface or the scripting language interffor processing.

  iv. **Process Management Tools:** this includes the execution management and monitoring. An instance of the business process can be started, stopped or paused via the execution management interface. The monitoring interface can retrieve status information for each process instance and load information for the entire system.

- **The WFMS Configuration server:** which is responsible for configuring and administrating the whole system, tuning the performance issues, and administrating user accounts and needs.

## CASE STUDY: CHECK VERIFICATION WORKFLOW SYSTEM

In a check verification business process, the main purpose is to examine checks received from customers and make payment according to each check or reject a check. Although the check verification process is usually made manually with high precision, the fast delivery of the service may in some situations placing a great burden on the organization and it results in some accidental mistakes. The process is best illustrated by the block diagram of Figure 4 and can be described in terms of the following activities and decision points:

i. The check is manually scanned and stored into an object database for further processing and archiving purposes.

ii. Image enhancement routines are executed on the scanned check image for eliminating the noise resulting from

the scanning process and removing unwanted piece of information.

iii. Feature extraction algorithms are applied to the check image to extract useful information such as the account number, the amount of payment, the customer name, signature, etc.

iv. The check is then indexed in a relational database based on some previously extracted accounting information.

v. The extracted signature from the check is subsequently analyzed by an automated Artificial Neural Network (ANN) classifier based on a database of signatures to verify the validity of the check signature.

vi. If the signature is invalid or the transaction is not acceptable, the check is rejected and returned to the customer.

vii. A task is then performed in order to determine if payment should be made to the customer.

viii. Validated checks are accepted with the corresponding transaction being performed and the payment is made to the customer.

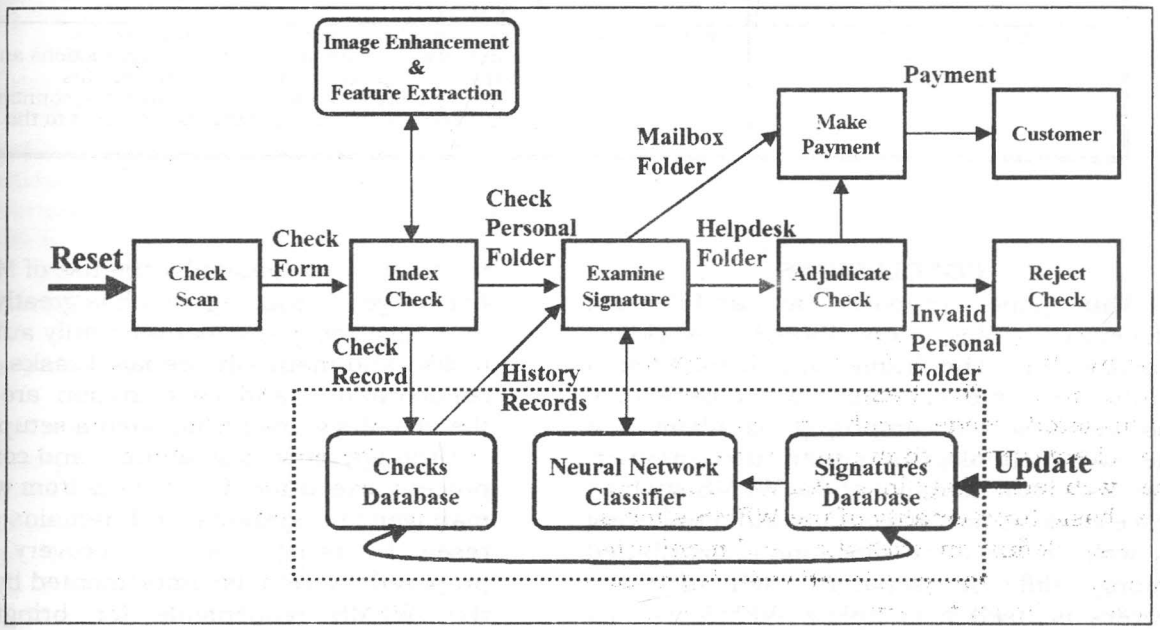


***Figure 4*** The check verification workflow system

In order to partially automate most of the tasks mentioned above for the check verification system, the Microsoft Exchange messaging tool is used. With this tool, a set of intelligent forms that replace paper forms are generated and used. These forms allow the user as well as external processes to easily enter, receive, post and route information in personal folders, public folders, or mailbox folders. Personal folders are the kind of storage areas for personal storage, backup, and design. Public folders contain folders to share among all users. Mailbox folders are password-protected folders for personal use.

Table 1 illustrates the kind of objects (forms or specialized folders) that have generated by the WFMS engine as part of the check verification system to facilitate the flow of information from manual and automated tasks. The objects have been generated to run under Windows NT using both Microsoft Exchange tools and Visual Basic language specifications.

***Table 1*** Generated objects for the check verification system

| OBJECT | TYPE | PURPOSE |
|---|---|---|
| 1. Scanned Check | Form | Lets the employee submits a form containing accounting information from the scanned check to the "Index Check" process. It can also be filled and prepared automatically by the "Check Scan" process. |
| 2. Check Signature | Personal Folder | The "Index Check" process uses image enhancement and feature extraction routines to get useful information about the check and the signature image. This information is then placed on the Check Signature Folder for further processing. |
| 3. Adjudicate Check | Helpdesk Folder | Checks with invalid signatures in which the automatic verification system failed to recognize are submitted to a special Helpdesk folder. A technician, whose job is to examine the signature manually for validity, subsequently accesses this folder. |
| 4. Invalid Payment | Personal Folder | The "examine signature" process generates information indicating that the check needs to be manually verified. The "Adjudicate Check" process will automatically fetch this folder and process its contents. |
| 5. Valid Payment | Mailbox Folder | The "examine signature" process generates information outlining the payment instructions and deduction transaction to be performed. This information is securely addressed to the accountant who is responsible for making the payment to the customer. |

## CONCLUSIONS

This paper proposed an architecture framework for distributed workflow management. The framework is based on a multi-tiered model combining the benefits of the client/server computing paradigm and the web technology in a distributed manner. The basic functionality of the WFMS such as process definition, execution and monitoring as well as resource allocation is distributed among different servers. Each of these servers is itself a complete WFMS with its own engine. Information concerning process execution is kept locally and there is no centralized server. The WFMS server that is close to the corresponding external applications can execute workflow activities. The WFMS components interact with each other using a message delivery protocol based on the well-known TCP/IP protocol and the Microsoft Exchange messaging tool. The proposed system is more reliable due to the fact that the failure of one or more WFMS server will not stop workflow process execution.

Although most real-world workflow systems such as the check verification system presented in this paper are difficult to be fully automated, the use of Microsoft Exchange messaging tool has greatly helped in filling the gaps between fully automated tasks and manually prepared tasks. Human collaboration and coordination are among the benefits gained from such a setup.

Although ensuring atomic and consistent process execution is missing from workflow management systems and remains an open research issue, failure recovery in the proposed system is implemented by having the WFMS responsible for bringing the process execution to a designated save point. The save point represents an acceptable intermediate state of process execution. To rollback workflow process execution, compensation activity will be invoked to undo the effects of the completed activities. Unlike database transactions which can be compensated and re-executed easily, we came to realize that workflow compensation is highly costly. It is therefore necessary to avoid unnecessary compensation whenever possible. In this research, we adopted static specification of compensation in which the workflow designer is himself responsible in determining the compensation to be performed in case of failure.

## REFERENCES


1. M. Rusinkiewicz and A. Sheth "Specification and Execution of Transactional Workflows," In Modern Database Systems: The Object Model Interoperability and Beyond. ACM Press and Addison Wesley, (1995).
2. K.D. Swenson and K. Irwin "Workflow Technology: Tradeoffs for Business Process Reengineering," In Proceedings of Conference on Organizational Computing Systems. ACM Press, pp. 123-129 August (1995).
3. C. Mohan, G. Alonso, R. Gnthr, and M. Kamath "Exotica: A Research Perspective of Workflow Management Systems," Data Engineering Bulletin, Vol. 18, No. 1, pp. 83-102, (1995).
4. J. Veijalainen, A. Lehtola, and O. Pihlajamaa "Research Issues in Workflow Systems," ERCIM Database Research Group Workshop on Database Issues and Infrastructure in Cooperative Information Systems, pp. 92-97 Norway, (1995).
5. W. Du and A. Elmagarmid "Workflow Management: State of the Art Vs. State of the Products," Hewlett-Packard Technical Report, HPL-97-90, (1997).
6. J. Tang and J. Veijalainen "Transaction-Oriented Workflow Concepts in Inter-Organization Environments," Proceedings of the International Conference on Information and Knowledge Management, pp. 187-192, Baltimore, (1995).
7. D. Worah and A. Sheth "Transactions in Transactional Workflows," Advanced Transaction Models and Architectures, Kluwer Academic Publisher, (1997).
8. M.C. Shan, J. Davis, W. Du, and Y. Huang "HP Workflow Research: Past, Present, and Future," Hewlett-Packard Technical Report, HPL-97-105, (1997).
9. M. Shan, J. Davis, W. Du, and Q. Chen "Business Process Flow Management and its Application in the Telecommunication Management Network," Hewlett-Packard Journal, Vol. 47 No. 5, pp. 86-92, (1996).
10. G. Alonso, R. Gnthr, M. Kamath, D. Agrawal, A. El Abbadi, and C. Mohan "Exotica/FMDC: A Workflow Management System for Mobile and Disconnected Clients," Distributed and Parallel Databases, Vol. 4, No. 3, pp. 31-38, (1996).
11. G. Alonso, M. Kamath, D. Agrawal, A. El Abbadi, R. Gnthr, and C. Mohan "Failure Handling in Large Scale Workflow Management Systems," IBM Research Report, RJ9913, (1994).
12. W. Du, M. Shan, and A. Elmagarmid "Consistent Execution of Workflow Processes," Hewlett-Packard Technical Report, HPL-97-71, (1997).
13. W. Du, G. Eddy, and M. Shan "Distributed Resource Management in Workflow Environments," Proceedings of the 5th International Conference on Database Systems Advanced Applications, pp. 165-171, Melbourne, Australia, April (1997).
14. J. Tang and J. Veijalainen "Transaction-Oriented Workflow Concepts in Inter-Organization Environments," International Conference on Information and Knowledge Management, pp. 217-223, Baltimore, (1995).
15. M. Kamath and K. Ramamritham "Correctness Issues in Workflow Manamement," Distributed System Engineering, Vol. 3, No. 4, pp. 51-59, (1996).
16. P. Attie, M. Aingh, M. Rusinkiewicz, and A. Sheth "Specification and enforcing Intertask Dependencies," VLDB Proceeding, Dublin, pp. 127-133, (1993).
17. D. Georgakopoulos and M. Hornick "A Framework for Enforceable Specification of Extended Models and Transactional Workflow," Journal of Intelligent and Cooperative Information Systems, Vol. 3 No. 3, pp. 34-39, (1994).
18. F. Leymann "Supporting Business Transactions Via Partial Recovery in Workflow Management Systems," BTW, (1995).


Received June 6. 1999
Accepted October 12. 1999

# إطار عمل تطبيقي لإدارة نظم تتبع الأعمال

عادل عبد المنعم الزغبي
قسم تكنولوجيا المعلومات – معهد الدراسات العليا والبحوث – جامعة الإسكندرية

## ملخص البحث

يهدف هذا البحث إلى تقديم إطار عمل لإعادة هيكلة وتصميم نظم المعلومات الإدارية والتي يتطلب تصميمها وتنفيذها كثيرا من الوقت والمجهود والخبرة وبالتالي تصبح عملية تعديلها لموائمة متطلبات مستجدة أو مستحدثة غاية في الصعوبة وهو ما يطلق عليه (Business Process Reengineering) ويقوم إطار العمل المقترح على نظام موزع لإدارة نظم تتبع الأعمال (Distributed Workflow Management) من خلال نظام ثلاثي الوحدات. الوحدة الأولى عبارة عن واجهة التطبيق التي يتعامل معها المصمم أو المستخدم للنظام والوحدة الثانية الوسطى تتكون من مجموعة خادمي التطبيقات المتنوعة التي قد يتطلبها التنفيذ أما الوحدة الثالثة فهي خلفية النظام الذي يخدم النظام ككل من خلال عملية تنفيذ التتبع المراد (Workflow). ويمتاز النظام المقترح بأنه يجمع بين فوائد نظام العميل / الخادم و وحدات خدمات الويب من خلال تقنيات موزعة. وقد تم اخذ مشكلة نظام تتبع التحقق من الشيكات وصرفها في البنوك كدراسة حالة لإظهار طريقة عمل الإطار المقترح و فوائده والمشاكل البحثية التي تم حلها بالفعل والمشاكل البحثية المعلقة إلى وقتنا هذا.