# Transaction in Document Approval Process Workflow

Yongliang Zhang
Tonghua 134002 China
e-mail: cnzhyl666@163.com

Huaiqing Ren
Tonghua 134002 China
e-mail: cnzhyl666@163.com

*Abstract*—Numerous transaction models have been proposed to mainly address the problems posed by advanced database applications in recent years, but only a few of these models are being used in real commercial products environments. We explore the complexity of workflow and show that, in many aspects, transaction models are a subset of workflow models and how traditional transaction processing can be incorporated to address some sophisticated problems up to date in document approval process workflow.

***Keywords- document approval; transaction model; workflow model; workflow management system***

## I. INTRODUCTION

Workflow technology continues to be subjected to on-going development in its traditional application areas of business process modeling and business process coordination, and now in emergent areas of component frameworks and inter-workflow, business-to-business interaction. A large number of workflow products, mainly workflow management systems(WFMS), are commercially available [1]. On the other hand, in the last decades, many transactions models have been proposed to address non-traditional applications. These models provide well-defined failure semantics in the sense of concurrency control and sophisticated recovery features through the ACID properties of database transaction theories[2].

Currently, there are several attempts to provide an execution platform flexible enough to support a variety of advanced transaction models. It is not clear, however, which are the relevant models and how they can be combined to construct a transactional workflows in an attempt to address more realistic environments. Much of the work is only transaction based, coupling advanced transaction technology and workflow management systems to support business processes with well defined failure semantics and recovery rules.

In this paper, we discuss the characteristics of workflow models taking Document Approval workflow(DAWF) as example. We show how workflow models have richer semantics and are more apt to be used in commercial products and provide a better perspective of the relationship between transaction models and workflow models, and proposed the rules for transaction models to be embedded into workflow process.

## II. DOCUMENT APPROVAL PROCESS WORKFLOW MODEL IN B/S MODE

A workflow management system is a software system that transforms the explicit representation of a workflow model into an internal and executable format and provides an operational environment for workflow model execution and the administration and monitoring of workflow model instances [3]. Document approval process workflow is a kind of simple workflow, the fig.1 demonstrates the document

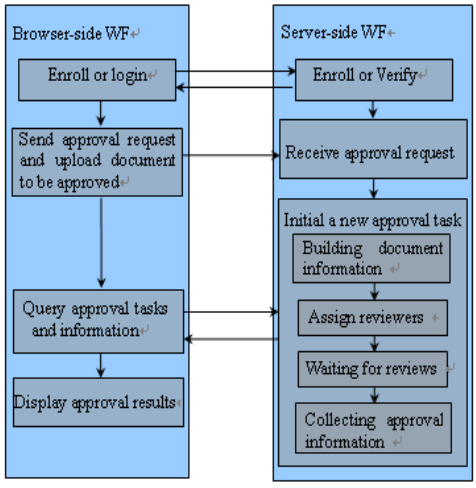


Figure1. The document approval process workflow model in B/S mode

978-0-7695-5004-6/13 $26.00 © 2013 IEEE
DOI 10.1109/ICCIS.2013.127

approval process workflow in B/S mode, a box represents a activity of the workflow, which can be nested also.

The vertical line with arrow is called control connectors between activities, is the order in which activities are executed, and the horizontal lines with arrow between host processes represent data communication process.

From the user's perspective, the approval process functions may be illustrated as follows:

1. A client enrolls to be a user of the approval process system.
2. After being verified, the user can participate in approval processes.
3. Once joined, a user can send a approval request and upload a document for approval using single, quorum or complex approval processes, and the user can predefine some reviewers through submitting necessary information.
4. A user may query and receive an approval task and respond to the request at any point in time.
5. The document information is presented in browser for review.
6. The user can approve or reject the document.

From the approval manage server application's point of view, the approval process functions can be illustrated as follows:

1. A user enrolls and requests to participate to the approval process system.
2. The approval manage server receives a request to be part of the approval process system.
3. The user information is stored in a database.
4. The approval manage server builds a new approval process and starting a new workflow instance.
5. Depending on the type of request (simple, quorum, or complex) a different activity is executed.
6. Every workflow is consisted of many activities, such as building document information, assign reviewers, and collecting approval information etc. the activities can be nested activities (as illustrated in fig.1).
7. Any user can query an authorized approval task and respond to the request at any point in time
8. The result of the approval process workflow is sent to the user.

## Ⅲ . TRANSACTION MODEL IMPLEMENTING IN DAWF

A WFMS automates the control flow and data flow between activities, and assign activities to users and programs. But existing transaction models limit themselves to only part of the problem, transaction models are only a subset of workflow. The most relevant feature provided by WFMS is their ability to describe an organization and coordinate the definition and execution of workflow processes to the particular characteristics of that organization. In a WFMS, the organization is described in terms of the roles, hierarchical levels and persons associated with it. A person can have several roles, manager, programmer, assistant, and a role can be assigned to several persons. When an activity is defined, the workflow designer must specify who is responsible for the execution of the activity. These activities do not necessarily happen automatically, as is assumed in advanced transaction models[4], but with direct user intervention. Even activities corresponding to programs that do not require human input for execution are associated with users who can monitor their progress and are responsible for their execution. The user can stop an activity, restart it, and force it to finish. Moreover, based on their ownership, users can initiate changes for modifying the deployment of all potential resources for performing their tasks, including application systems or workflow models. Structural changes may affect a number of participants. Only process owners can make structure changes to workflow models accessible to them, either making new versions or changing running instances of the models. A major difference between WFMS and transaction models is in the area of correctness and reliability. Current WFMS do not offer significant support for recovery and failure handling. In most cases, user intervention is required, either to solve consistency problems or to specify which activities are needed to recover from an exception. All of the above are not implementing in WFMS in the form of transaction models. Many WFMS log this information in an execution history, which is kept for each workflow instance when tracking the execution of a workflow instance or when rolling back workflow instances in case of failures [5]. History can not be inversed, just like a raw material has been manufactured into products, which can not be return to its original form; as a result, some workflow can not be implemented in the form of transactions.

We propose a flexible transaction scheme for document approval process workflow, Flexible transactions work in the context of flexible database environments. There are two kinds of transactions, one is traditional transaction whose effect can be undone, the others can not be undone, but can be rollback and retry again, or an alternative sub-transaction can be available. Correctness is guaranteed by enforcing certain rules in the order of execution of the sub-transactions and the overall structure of the flexible transaction. A flexible transaction is well formed when the possible orders of execution do not violate the data dependencies between sub-transactions and the transaction is atomic. Generally, we make the following rules to implement transactions in DAWF

1. Each transaction corresponds to an activity, and a task corresponds to an activity also, which is consisted of many more nested sub-transactions, and the nested level can be unlimited.
2. The order of activities follows the ordering of the corresponding transactions, there can be no sequence for some distributed transactions, e.g. a quorum approval of a document by several reviewers is no sequence for these reviewers, who approve first is nothing important.
3. The transaction composed of sub-transactions that have no sequence in order can be partially done a rollback, that is, only a rollback is done for a certain sub-transaction, others are remained without any effect regardless the order of execution
4. Any activities have an exit condition that evaluates to true only when all the sub-transactions in it execute successfully, otherwise, the exit condition evaluates to false.
5. When an activity is composed of sub-transactions that all have no sequence in order, abort of a sub-transaction can not stop the committing of other sub-transactions.
6. An optional alternative path consist of compensable activities can be added to the path of the sub-transaction which can not be undone, when the sub-transaction commit, and there is some mistake, the sub-transaction has to be aborted, but the effect can not be undone, then the compensable activities in this path will be executed automatically according to output evaluation.
7. When an activity terminates, the status of the corresponding sub-transaction, committed or aborted, is recorded in the output data list of the block. This data list will be mapped into the input data list of the corresponding compensating block.
8. Every activity has a precondition as a part of input, the precondition maybe the results of the execution of the previous activities. We introduce a flexible activity for each block of compensating activities connecting to the compensating activities with a control connector in which the transition condition is that the relevant activities have been committed. The connectors between the compensating activities are the same as those for the corresponding compensable activities but reversed. Information about activities being executed and aborted can be found in the input data list of the block.
9. Abort an execution path to restart from another is done by compensating all the activities committed along the alternative path. Note that there is always an appropriate point for any sub-transaction to rollback to, and at this point the flow will be redirected to the corresponding compensating activities path, and the execution of the new path will be started.

## Ⅳ. DATABASE DESIGN

Iorder to realize the process, we design database Entity Relationship(E-R) just showed in figure 2, according to the E-R chart, realize personnel, departments, task, and process management. and process control by I(I=1,2,3,4,5 etc). task (document) management.

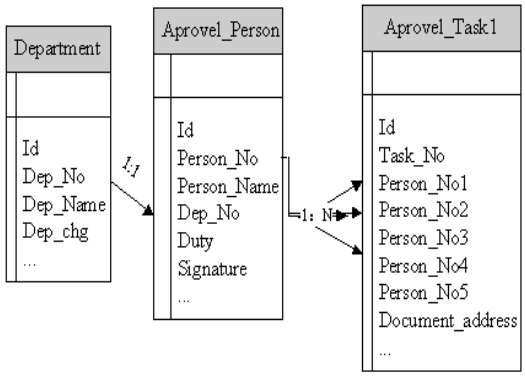


Figure2. Entity-Relation chart

According to the approval process design and database technology, we can use of database technology to finish the examination and approval of the business process automation management.

## Ⅴ. CONCLUSIONS

We have discussed the characteristics workflow and present the idea that the transaction model is an important

aspect of workflow, document approval workflow in B/S mode is illustrated in details, and the transaction model implementing methods in document approval workflow is given in this paper.

## ACKNOWLEDGEMENTS

The authors would like to thank the support of the project of Education Department of Jilin Province of China NO. [2011] 313

## REFERENCES

[1] Van Der Aalst, M. Wil, Arthur H. and P. Alistair, "Workflow Patterns", Distributed and Parallel Databases, 2003, 14(1): pp. 5-51.

[2] A. Sheth and Rusinkiewicz, "On Transactional Workflows", Bulletin of the Technical Committee on Data Engineering, 1993,16(2), pp.23-59.

[3] C. Bussler, "Specifying Enterprise Processes with Workflow Modeling Languages", In Concurrent Engineering: Research and Applications. Special Issue: Application of Enterprise Modeling Languages for Concurrent Engineering. Vol. 4, No. 3, September 1996

[4] W.M.P. van der Aalst and K.M. van Hee, "Workflow Management: Models Methods and Systems, MIT press, Cambridge, MA, 2002.

[5] F. Leymann and D. Roller, Production Workflow, Prentice Hall, 2000.