Formal description and compensation implementation of workflow transaction

Zou Jinan

Department of Computer, Putian University, Putian City, China

ABSTRACT

Workflow transaction is one of the most difficult workflow problems which should be solved as soon as possible. Most of the existing transaction models describe complex workflow semantics with an informal specification, thus limits their applicability in complex application scenarios. By relating formal specification and practical application of workflow environments, the paper presents an approach to long transaction management in workflow environments, the approach is based on the well-known notion of compensation, and it extends to deal with both arbitrary process structures to allow cycles in processes and through consistent points to allow partial compensation of processes. The paper also presents a formal description of the transaction model and transaction management algorithms in set and graph theory, provides clear and unambiguous transaction semantics.

Key words: Workflow transaction management; long transaction; consistent point; transaction compensation

INTRODUCTION

Workflow technology has become a hotspot of many experts in the field, as one of the most difficult problems of the workflow technology, the study on workflow transaction has been put on the agenda. However many existing transaction models adopt informal way to describe the transaction semantics, this leads to great limit of their usability in complex scenes. The paper studies on how to link the formal description to the workflow system application, the transaction model which generated by the compensation mechanism has a low limit of transaction attribute and rollback. The transaction model is based on the saga model[1] and it adds the support of ring structure. Part of the compensation is also introduced to realize the transaction consistent point[2]. Because of many workflows in application includes the ring process structure such as with iterative way to achieve business objectives or retry a business function etc., so it is very important to add to support of the ring structure. The importance of the partial transaction compensation of the model is that it can provide a more flexible way of controlling process rollback. This paper gives a formal description of transaction management algorithm and
Transaction model with the set theory and graph theory, defines a clearly transaction model of operational semantics, describes the mapping relationship from formal function to a modular system structure.

**TRANSACTION DESCRIPTION AND COMPENSATION ALGORITHM**

Transaction is used to deal with the higher layer business process (long time operation) and low layer business process (short time operation) for different needs. The above two kinds of processes exist in most applications, and the low layer is the sub-process of the higher. Workflow management needs a long time operation processing, high-level cooperative transaction model, FLTM (Flexible Long Transaction Model) model[3] is designed for this purpose.

Local transaction or short transaction is the one which has atomic action in workflow process, the FLTM’s local follow strictly the ACID characteristics, global transaction or long transaction follows the isolation of relaxation. By allowing a transaction context access to intermediate transaction results to improve concurrency and execution efficiency. In order to avoid the cancellation of all works that has been executed and share the intermediate result, global transaction rollback operation should contain specific application semantics. For this purpose the concept of partial compensation is introduced. The rollback of global transaction is implemented through the compensation of executing global transaction. Global transaction is composed by the compensation of a series of local transactions.

In the failure of operation the long transaction, short transactions which have not been submitted can be terminated, those have been submitted would be compensated. Compensation operation which relates to the application needs to be specified by the application designer. In order to realize part of compensation, it is necessary to introduce the concept of consistent point which provides more flexible partial compensation strategy when failure in processing global transaction. Different from the security point of Saga model, consistent point is completely set dependent on the special semantic application. It is unnecessary to do inspection just like the checkpoint in database.

**GLOBAL TRANSACTION DESCRIPTION GRAPH AND EXECUTION**

Global transaction description graph is a directed graph that consists of a series of process elements and ordered elements. Process element is used to represent the node of the long transaction description graph, there are two types of the node: one is the task node which represents the execution of activities and the other is the logical node which used to construct a complicated routing structure. A task node is an entity that has short transaction properties, it can be defined as a triple: 

\[ Tspecific = < task, comp, consistent > \]

where task is the task descriptor, comp is the compensation task descriptor, consistent identifies if this task is a consistent point.. The logical node is used to represent various branches of LSspecific and convergences of LJspecific, that is: 

\[ Lspecific = LSspecific \cup LJspecific \]

where total identifies if it is an and logical node or an or logical node. Based on these basic types, there are a variety of complex logic nodes, for example, iterative branch node, convergence node and part convergence node[4]. To sum up, the process of element Vspecific is defined as: 

\[ Vspecific = Tspecific \cup Lspecific \]

The ordered element is used to describe the arc of graph, it is defined as: 

\[ Especific = < orig, dest, cond > \]

where orig and dest represent the starting point and end point of arc, cond is used to choose the path in the or branch, TRUE is its default value in other cases. Global transaction description graph is consisted of a series of process elements and ordered elements., that is 

\[ Gspecific = Vspecific \cup Especific \]

The starting point and end point of description graph are defined as following:

\[
\text{start}(G) = \{ v \in G.V \mid \neg( (w \in G.V) \land (w, v) \in G.E) \} \\
\text{end}(G) = \{ v \in G.V \mid \neg( (w \in G.V) \land (v, w) \in G.E) \}
\]

The right description graph has only one starting point and at least one end point, this constraint is expressed as:
The function card is used to calculate the cardinality of a set. If function predv and succvAre are used to calculate the direct precursor node and successor node of vertex v in graph G, then:

\[
\text{predv}(G, v) = \{ w \in G.V | (w, v) \in G.E \}
\]

\[
\text{succv}(G, v) = \{ w \in G.V | (v, w) \in G.E \}.
\]

Global transaction execution takes the transaction description graph as its input, and incrementally generates execution graph of long transaction execution history. The emphasis is execution of graph definition, the basic predicates and functions and the construct function of modifying the execution graph. The following introduces short transactions which constitute a long transaction as beginning. Instances of short transaction constitute the basic steps of long transaction. For simplicity, the following uses the "short transaction" to represent "short transaction instance", and Tloc represents a collection of short transaction identifier. There are two predicates which named started and committed defined in the short transaction, they are used to represent respectively two execution state whether the transaction has begun or submitted. Consistent, dummy, idempotent[4], alternative are also predicates.

Consistent is used to detect of if the short transaction is a consistent point, dummy is used to detect if the execution of the short transaction has no effect, idempotent is used to detect if the result of several time is same as the single time one, alternative is defined as when the class activities can not be successfully executed the it can perform alternative activities to achieve the same effect. Partial compensation only needs to compensate the execution process from rollback point to the consistent point[5]. The main problem of partial compensation is to find suitable sub-graph. When searching there need to consider consistent point, task dependencies in the forward recovery and backward recovery. Following is some algorithms of calculating sub-graph.

**IMPLEMENTATION MECHANISM**

Fig. 1 shows the interaction of LTSS and workflow system. The left part of the figure is LTSS, the right part is the workflow system that using LTSS, the bottom of the figure is the persistent storage used to store information of long transaction description and execution graph.

The core of the workflow system is the process engine and its maintenance of process instances, it is also responsible for the interpretation of the workflow process description graph and scheduling the process instances. The process instance represents one call to process description graph, and the process engine controls the description graph through interface of them.

The core of LTSS is LT engine which is responsible for the maintenance of LT instances. One LT instance represents one running long transaction instance and it saves and maintains long transaction execution graph and other relevant information. Process instance and LT instance are dynamically created and deleted, the two are mutually corresponding. The process informs the LT instance about process related events through interface and then updates the state of long transaction. When terminating long transaction, the process engine will inform LT engine, then LT engine will search the long transaction execution graph and calculate the required compensation graph, after finishing it will inform LT instance for saving the transaction description, then the process engine will compensate the name of transaction and its restart point.
FORMAL DESCRIPTION OF TRANSACTION SEMANTIC

Over the past ten years, the advanced transaction model (ATM) for solving problem of long transaction has attracted a lot of attention. The classical ATMs include nested transactions[6], multilevel transactions[7] and sagas[1]. The ATCA[8] frame gives the concept frame of constructing, analyzing and extending transaction model. Contract[10] provides reliable execution environment for long time calculation, It is used to implement transactional workflow. Although ATM solves the long transaction problem from different levels and aspects, it is still based on the data center, while workflow is based on the process center, so the characters provided by ATM is only a subset of the characters which workflow system needs.

The semantic and support of high level transaction were described above, in this level the sagas rollback was implemented by compensation[1]. In comparison, compensation method of FLTM is more comprehensive, flexible: the separation of description graph and the execution graph makes it is more naturally in the process of ring structure, the persistent point provides flexible partial compensation, the filtering of compensation graph reduces the compensation price as far as possible. Now compensation has become an effective method for treatment of rollback application. The reference [9] discussed the affairs of the compensation mechanism of multilevel transaction model, its emphasis is to determine the scope of compensation in the nested structure while the paper focuses on the structure and compensation of any process graph. ConTracts[10] is designated to step group compensation in accordance with the compensation block, while the compensation graph in the paper is a dramatic one. Due to the strict definition of consistency, partial compensation in ConTracts must be careful to use; partial compensation of this article is the default method and easy to implement.

The reference [11] gives the formal description of transaction compensation, its focus is the correctness of each transaction compensation, while the paper’s is to construct complex compensation graph according to compensation transaction predefined, so the paper is a supplement to the reference [11]. The reference [12] gives the formal method of complex transaction in complex system, it describes a complex business environment which is combination of three basic structure(stack, branch and sink). The above framework analyzes the concurrency characteristics of sagas in the advanced abstraction layer, but it has not analyzed the operation semantic of sagas, so the paper can also be taken as the specific content of the reference [12].
CONCLUSION

The paper’s work mainly displays in the following three aspects: first of all, it extends the well-known saga model to process complex structure and partial compensation and gets a transaction model which has been a practical application in the workflow environment. Secondly, it gives a formal description and transaction management algorithms of transaction model in a simple way, it also gives the complete operational semantics of transaction management of the model. Combination of the formal description and operation methods make it is more easy to implement the advanced transaction mechanism. Formal semantics given in the paper is useful not only for the users of developing the transaction management system but also for the advance users of low level transaction model. It lays the foundation for static testing of transactional behavior of complex workflow process. The compensation algorithm can be taken as a tool of analyzing workflow, it can provide automatic help for designer and the terminal users who deal with long transaction design. Third, realization of this mechanism can be integrated into the commercial workflow management system which is module and loosely coupled. In a word, this paper gives a complete workflow transaction management framework from advanced description to real implementation, fills the gap between specification and implementation. The prototype system has been implemented in the transaction mechanism, the future work is to extend the scheme to distributed and multi-engine environment.

Acknowledgments
The authors wish to thank the Major Project supported by Science and Technology Department of Fujian Province of China for contract 2012H1009, under which the present work was possible.

REFERENCES