Extended Transaction Models and the ACTA Framework

Panos K. Chrysanthis\textsuperscript{1} and Krithi Ramamritham\textsuperscript{2}

\textsuperscript{1}Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, USA
\textsuperscript{2}IIT Bombay, Powai Mumbai, India

Synonyms

Advanced transaction models; Generalization of ACID properties

Definition

Although powerful, the transaction model adopted in traditional database systems is found lacking in functionality and performance when used for applications that involve reactive (endless), open-ended (long-lived) and collaborative (interactive) activities. Hence, various extensions to the traditional model have been proposed, referred to as extended transactions. These models are characterized by the structure of their transactions, the commit and abort dependencies and the visibility rules among transactions. ACTA is a comprehensive transaction framework that facilitate the specification, analysis and synthesis of extended transaction models. The name ACTA, meaning actions in Latin, was chosen given the framework’s appropriateness for expressing the properties of actions used to compose a transactional computation.

Key Points

By means of the notion of transactions, database systems offer reliability guarantees concerning the correctness of data in spite of failures and concurrent accesses by multiple users. However, the transaction model as well as the simple data model adopted in traditional database systems have been found lacking in functionality and performance in their support of the emerging advanced database applications such as design databases, computer publishing, network management, multidatabases and mobile databases. In order to deal with the inherent limitations of the traditional data and atomic transaction model, researchers have proposed semantic and object-oriented data models and extensions to the traditional transaction model. Nested transactions was the first such extension that added a hierarchical structure to the traditional flat atomic transactions. The hierarchical structure allows concurrency within a transaction and fine-grained failure and exception handling since sub transactions can abort independently without causing the abortion of the whole transaction.

The original nested transaction model was subsequently enhanced with new types of sub transactions, relaxed abort and commit dependencies and visibility rules for externalizing partial...
results among transactions. These extensions led
to a variety of open-nested transactions models
such as Sagas, Split Transactions, Flex Trans-
actions, ConTracts and S-transactions, and of
correctness criteria such as quasi serializability,
epsilon-serializability, semantic atomicity, quasi
failure-atomicity.

All the above extensions have been introduced
with specific applications or with specific trans-
action properties in mind [2]. Their ad hoc char-
acter makes it difficult to identify the properties
of transactions that adhere to a particular model
and to ascertain in what respects an extended
transaction model is similar or different from
another. The need for a comprehensive trans-
action framework that would facilitate the pre-
cise specification of the properties of a model,
vis a vis visibility, consistency, recovery and
permanence, and allow the formal comparison
different models led to the development of
ACTA [1]. ACTA is a first-order logic based
formalism with a precedence relation that al-
 lows a transaction modeler to specify both the
high level properties (requirements) of a model
and the lower level behavioral aspects of the
model in terms of axioms. Specifications include
the following four components: (i) the set of
transaction management events associated with
the transaction model, such as begin, commit,
abort, split, and join; (ii) the semantics of these
significant events, characterized in terms of their
effect on objects (their value and synchronization
state) and other transactions (different types of
dependencies, such as commit dependency and
abort dependency); (iii) the view of each transac-
tion, specifying the state of objects visible to that
transaction; and (iv) the conflict set of each trans-
action, containing those operations with respect
to which conflicts need to be considered.

Besides supporting the specification and anal-
ysis of existing transaction models, ACTA has
the power to specify the requirements of new
transactional applications and synthesize models
that satisfy these requirements. This was demon-
strated by deriving new transaction definitions ei-
ther by starting from first principles or by modify-
and/or combining the specifications of existing
transaction models. The exercise of analyzing

Cross-References

- ACID Properties
- Compensating Transactions
- ConTract
- Correctness Criteria beyond Serializability
- e-Commerce Transactions
- Flex Transactions
- Generalization of ACID Properties
- Internet Transactions
- Multilevel Transactions and Object-Model
  Transactions
- Nested Transaction Models
- Open Nested Transaction Models
- Polytransactions
- Sagas
- Semantic Atomicity
- Split Transactions
- Transaction
- Transaction Management
- Transactional Processes
- Workflow Transactions

Recommended Reading

1. Chrysanthis PK, Ramamritham K. Synthesis of
extended transaction models using ACTA. ACM Trans
2. Elamagarmid AK. Database transaction models for
advanced applications. Los Altos: Morgan Kaufmann;
3. Ramamritham K, Chrysanthis PK. A taxonomy of
correctness criteria in database applications. VLDB J.