Volume 3, No. 2, March-April 2012

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Performance Evaluation of Mobile Transaction Models

Veenu Saini
Assistant Professor, Department of IT
PG Govt. College-11, Chandigarh, India.
vnusaini@yahoo.co.in

Abstract: In order to support the development of mobile transaction processing system, there exist some requirements that a mobile transaction processing system must face. These requirements not only focus on customizing the transaction properties, but also take into account other challenging characteristics of mobile transactions such as mobility of transactions, heterogeneity, disconnected and distributed transaction processing. There are many mobile transaction models, analyzing tools and transaction processing systems that have been proposed and developed to support mobile transaction processing. However, there are still major limitations, especially to support both the disconnected processing and the mobility of transactions. In this paper I provide performance evaluation of existing mobile transaction model with the requirements of transaction processing system that are ACID properties, mobility, disconnected and distributed execution and heterogeneity. I believe that these results have significant implications for designing or proposing a mobile transaction model.

Keywords: transaction processing, ACID properties, mobility, heterogeneity

I. INTRODUCTION

Due to the nature of the mobile computing environments, transaction management has to be reevaluated for mobile databases. The transactions in mobile computing environments are usually long-living transactions, possibly covering one or more disconnected durations. Supporting disconnected operation (i.e. allowing a mobile host to update autonomously during disconnection) raises issues in consistency. Providing disconnected operation also requires some pre-caching of data that will be required for the necessary operations to be performed during disconnection.

The ACID (atomicity, consistency, isolation and durability) properties of a transaction ensure that: (a) a transaction always keep the database in a consistent state, (b) a transaction does not disturb other transactions during their concurrent execution processes, and (c) the consistent state of the database system that is established by a committed transaction withstands software or hardware failures.

There are many new transaction models [1] [2] that have been developed to support transactions in mobile environments. One common approach is to provide for the transaction processing systems adaptability to deal with different environment conditions and to cope with the constraints of mobile computing resources. However, there are still several major limitations. For example, the architecture of mobile transaction environments [3] relies too much on the mobile support stations. The ability to support both the disconnection and mobility is still a major challenge for mobile transaction models.

In this paper, I have evaluated the performance of existing mobile transaction models (Kangaroo transaction model, Report and Co-transaction model, Two-tier transaction model, Pro-motion transaction model, Weak - Strict transactions model, Pre-serialization transaction model and Moflex transaction model) on five main requirements so that researcher can design a new model by (a) Improving the disconnected transaction processing, (b) Capturing the mobility of transactions in mobile environments, (c) Supporting the distributed execution, (d) Ensuring the ACID Properties, and (e) supporting heterogeneous database.

II. MOBILE TRANSACTION MODELS

A. *Kangaroo Transaction Model (KTM):*

a. *Description:*

The Kangaroo transaction model [4] [5] is designed to capture the movement behavior and the data behavior of transactions when a mobile host moves from one mobile cell to another. This transaction model is built based on the concepts of global and split transactions in a heterogeneous and multi-database environment. The global transaction is split when the mobile host moves from one mobile cell to another, and the split transactions are not joined back to the global transaction. The Kangaroo transaction model assumes that the mobile transactions may start and end at different locations. The characteristics of the Kangaroo transaction model are

a. Mobile transactions that include a set of sub-transactions called global and local transactions are initiated by mobile hosts. These mobile transactions are entirely executed at the local database servers that reside on the fixed and wired connected networks.
b. The movement of the mobile host from one mobile cell to another is captured by the splitting of the on-going Joey transaction at the old mobile support station and the creating of new Joey transaction at the new mobile support station. The execution of the Joey transaction is supported by the Data Access Agents (DAA) that act as the mobile transaction managers at the mobile support stations.

b. *Transaction Properties:*

The Kangaroo transaction is the basic unit of computation in mobile environments. The serializability of mobile transactions is not guaranteed, and there is no dependency among Joey transactions, i.e., each Joey transaction can commit independently. Two transaction processing modes, which are *compensating* and *split* modes, are supported by the model. For compensating mode, when a failure occurs, the entire Kangaroo transaction is undone by executing compensating transactions for all those Joey transactions. For split mode, the local DBMS takes responsibility for aborting or committing sub-transactions.

c. Mobility:

The Kangaroo transaction model keeps track of the movement of mobile hosts via the support of the DAA that operates at the mobile support station. In other words, the mobility of mobile hosts is captured on the condition that the mobile hosts always may communicate with the mobile support stations. While mobile hosts move from one mobile cell to another, the hand-off processes are carried out by the DAAs.

d. Disconnection:

Disconnected transaction processing is not considered in Kangaroo transaction model. The processing of Kangaroo transactions is entirely moved to the fixed database servers for executing.

e. Distributed Execution:

The mobile transactions are initiated at the mobile hosts, and entirely executed at fixed hosts. Transaction results are forwarded back to the mobile hosts. The Kangaroo transaction model has shown that the structure of mobile transactions at the specification and execution phases (with the dynamic support of Joey transactions) can be different because of the mobility behavior, i.e., fast or slow movements, of the mobile host.

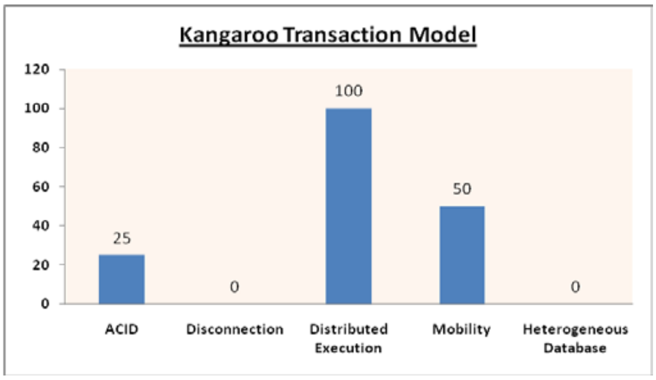

Figure 1: Bar Chart of Kangaroo Transaction Model

Figure 1 shows that in KTM, atomicity is ensured but it does not assure consistency, durability & isolation property. KTM does not support disconnected transaction processing. KTM supports the distributed execution fully and mobility partially.

B. Reporting and Co-transaction model (RCTM):

a. Description:

Reporting and Co-transactions transaction model [1] is based on a two level nested transaction model. A reporting transaction *TR* shares its partial results to top-level transaction *S* by delegating its operations. The delegation process can happen at any time during the execution of transaction *TR*. A co-transaction is a reporting transaction but it cannot continue executing during the delegation process. Thus, the co-transaction behaves as a co-routine, and resumes execution when the delegation process is completed.

This model arranges the mobile transaction into following four types:

Atomic transactions: It is related with substantial events the normal aborts and commits properties.

Non-compostable transactions: It is not linked with compensating transaction. It can execute at any time and the parents of these transactions have the responsibility to commit and abort [6].

Reporting transactions: A report can be regarded as a delegation of state between transactions. The reporting transaction not assigning all its results to its parent transactions .It only has one receiver at any time during execution. The updating is completed permanently if receiving parent transaction is successfully executed but if receiver parent transactions unsuccessfully terminate then corresponding reporting transaction abort.

Co-transactions: These transactions executed like co-procedures executed. When one transaction is executed then control passes from current transaction to another transaction during sharing the results. At a time either both transaction successfully executed or failed.

b. Transaction Properties:

The top-level transaction is the unit of control, and atomic sub transactions are compensable transactions. A Reporting transaction that is compensable does not have to delegate all of the committed results to the top-level transaction when it commits. Sub-transactions that are non-compensable delegate all of their operations to the top-level transaction when it commits.

c. Mobility:

The locations of mobile hosts are determined via the identification of mobile support stations. However, the model does not mention explicitly what happens when mobile hosts move from one mobile cell to another

d. Disconnection

Delegation operations require a tight connectivity between the delegator (i.e., Report and Co-transaction) transactions and the delegate transaction (i.e., the top level transaction). Therefore, disconnection is not supported in this model.

e. Distributed Execution:

The model supports distributed transaction processing among mobile hosts and fixed hosts where the network connectivity among these hosts is assumed to be available when it is needed.

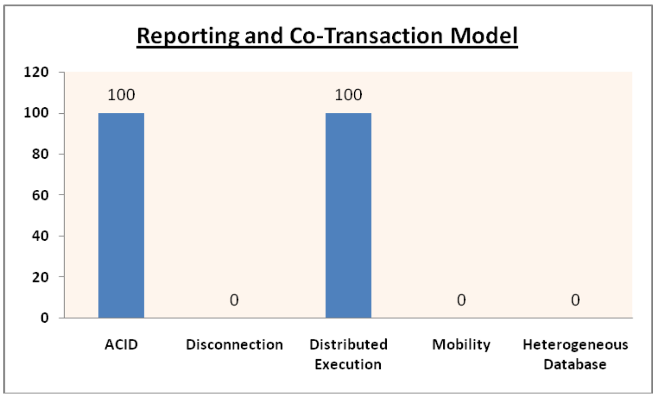

Figure 2: Bar Chart of Reporting and Co-Transaction Model

As shown in Figure 2, RCTM fully supports the transaction properties (ACID) and distributed execution. Disconnection, mobility and heterogeneity are not supported by RCTM.

C. *Pro-Motion Transaction Model (PMTM):*

a. *Description:*

The Pro-motion transaction model [4] is a nested transaction model. The Pro-motion model focuses on supporting disconnected transaction processing based on the client-server architecture [7]. Mobile transactions are considered as long and nested transactions where the top-level transaction is executed at fixed hosts, and sub transactions are executed at mobile hosts. The execution of sub-transactions at mobile hosts is supported by the concept of compact objects.

Compact objects are constructed by compact manager at database servers. Necessary information is encapsulated within a compact object. The compact objects are co-managed by the compact managers (resided at the database servers), the mobility managers (at base host), and the compact agents (at the mobile hosts). The compact object plays a role as a contractor [8] that supports data replication and consistency between mobile hosts and database servers. When a mobile host is disconnected, the compact agent takes responsibility for managing all local database operations of mobile transactions at the mobile host. When the mobile host reconnects to database servers, the compact objects are verified against global consistency rules before the locally committed mobile transactions are allowed to commit. Transaction processing consists of four phases: hoarding, disconnected, connected, and resynchronization [9]. Shared data is downloaded to the mobile host in the hoarding phase. When the mobile host is disconnected from the fixed host, transactions are disconnectedly executed at the mobile host. If the mobile host connects to the fixed database, the transactions are carried out with the support of the compact manager. When the mobile host reconnects to a fixed host, the results of local transactions are synchronized with the database.

b. *Transaction Properties:*

The Pro-motion transaction model supports ten different levels of isolation. Transactions are allowed to locally commit at mobile hosts; the committee results of these transactions are made available to other local transactions. However, the local committed results must be validated when the mobile hosts reconnect to the database servers. Therefore, the durability property of transaction is only ensured when the transaction results are finally reconciled at the fixed database.

c. *Mobility:*

Though the mobility manager supports communications between the mobile host and the database servers, however in the Pro-motion transaction model the feature of mobility is not explicitly discussed.

d. *Disconnection:*

Pro-motion transaction model supports disconnected transaction processing via the support of compact objects. When the mobile host is disconnected from the fixed database, the sub-transactions are split and executed at the mobile host (these split sub-transactions are not joined when the mobile host reconnects to the fixed database). Disconnected transaction processing is a dominant transaction processing mode in Pro-motion even when the mobile hosts are able to connect to the database server. Therefore, the Pro-motion transaction model requires high-capacity mobile resources at the mobile hosts.

e. *Distributed Execution:*

Transactions are mostly executed at mobile hosts and the results are reconciled at the database servers. Therefore, the distributed transaction processing is not strongly supported by the model.

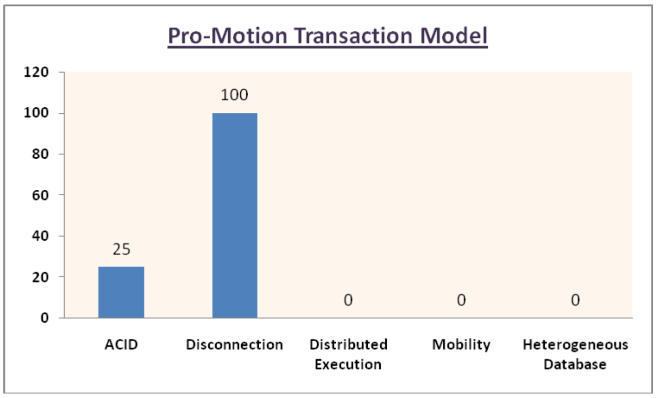

Figure 3: Bar Chart of Pro-Motion Transaction Model

Since only durability is ensured in PMTM, ACID is 25% only and disconnection is fully supported.

D. *Two - Tier Transaction Model (2TTM):*

a. *Description:*

The two-tier (also called Base-Tentative) transaction model is based on a data replication scheme. For each data object, there is a master copy and several replicated copies. There are two types of transaction: Base and Tentative. Base transactions operate on the master copy; while tentative transactions access the replicated copy version. A mobile host can cache either the master or the copy versions of data objects. While the mobile host is disconnected, tentative transactions update replicated versions. When the mobile host reconnects to the database servers, tentative transactions are converted to base transactions that are re-executed on the master copy. If a base transaction does not fulfill an acceptable correctness criterion (which is specified by the application), the associated tentative transaction is aborted.

b. *Transaction Properties:*

Tentative transactions locally commit at the mobile host on replicated copies, and the committed results are made visible to other tentative transactions at that mobile host. The final commitments of those tentative transactions are performed at the database servers.

c. *Mobility:*

Two-tier transaction model does not support the mobility of transactions.

d. *Disconnection:*

While the mobile hosts are disconnected from the database servers, tentative transactions are locally carried out based on replicated versions of data objects.

e. Distributed Execution:

Two distinct transaction execution modes are supported: connected and disconnected. Transactions are tentatively carried out at disconnected mobile hosts, and re-executed as base transactions at the database servers.

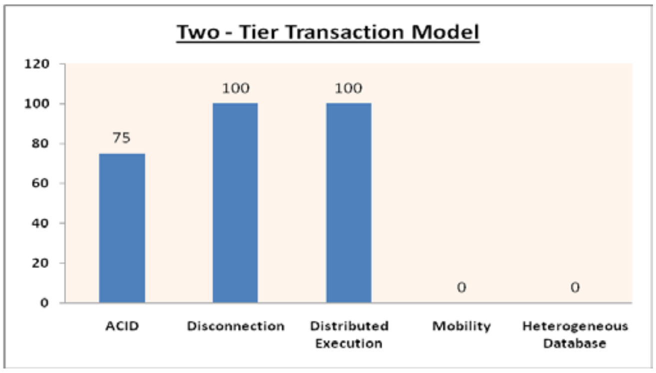

Figure 4: Bar Chart of Two-Tier Transaction Model

2TTM does not support isolation, mobility and heterogeneity

E. Weak-Strict transaction model (WSTM):

a. Description:

The Weak-Strict (also called Clustering) transaction model consists of two types of transaction: weak (or loose) and strict [7]. These transactions are carried out within the clusters that are the collection of connected hosts which are connected via high-speed and reliable networks [10]. In each cluster, data that is semantically related is locally replicated. There are two types of a replicated copy: local consistency (weak) and global consistency (strict). The weak copy is used when mobile hosts are disconnected or connected via a slow and unreliable network. Weak and Strict transactions access weak and strict data copies, respectively. When mobile hosts reconnect to database servers, a synchronization process reconciles the changes of the local data version with the global data version.

b. Transaction Properties:

Weak transactions are allowed to commit within its cluster, and results are made available to other local weak transactions. When mobile hosts are reconnected, the results of weak transactions are reconciled with the results of strict transactions. If the results of a weak transaction do not conflict with the updates of strict transactions, weak transactions are globally committed; otherwise they are aborted.

c. Mobility:

The concept of transaction migration is proposed to support the mobility of transactions, and to reduce the communication cost. When the mobile host moves and connects to a new mobile support station, parts of the transaction that are executed at the old mobile support stations are moved to the new one. However, no further details about the design or implementation are given.

d. Disconnection:

The Weak-Strict transaction model supports transaction processing in disconnected and weakly connected modes via weak transactions.

e. Distributed Execution:

Transaction execution processes can be distributed between the mobile host and the database servers within a cluster that the mobile host participates in. However, the distributed transaction processing among mobile hosts in a cluster is not discussed.

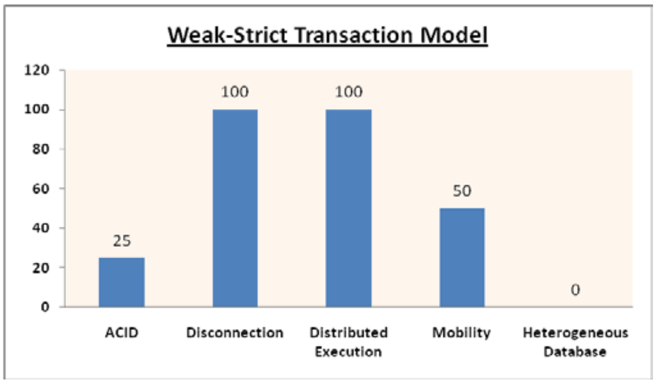

Figure 5: Bar Chart of Weak-Strict Transaction Model

In transaction properties only consistency is supported by WSTM and mobility is partially supported by WSTM.

F. Pre-serialization transaction model (PSTM):

a. Description:

Pre-serialization transaction model [5] is built on top of local database systems. Mobile transactions (also called global transactions) are submitted from mobile hosts through the global transaction coordinators that reside at the mobile support stations. The mobile transaction is entirely processed at local database systems. At each node (or site), there is a site manager that administrates all the transactions executed at that node. When a global transaction is prepared to commit, a global transaction coordinator will carry out an algorithm, called Partial Global Serialization Graph algorithm that detects any non-serializable schedule among the mobile transactions. If there is a cycle in the graph, i.e., the schedule is non-serializable, the mobile transaction is aborted.

b. Transaction Properties:

Each sub-transaction of a global transaction is managed by the local transaction manager. The global serializable graph of transactions is constructed by collecting sub-graphs from the local sites. The atomicity property of the global transaction is relaxed by the concepts of vital and non-vital sub-transactions.

If a vital sub-transaction aborts, its parent transaction must abort. However, the parent transaction does not abort if a non-vital sub-transaction aborts. When a sub-transaction commits at the local database system, the results are made visible to other transactions at this local database system.

c. Mobility:

The global transaction coordinators that reside at the mobile support stations support the mobility of mobile transactions. This is done by transferring the global data structure from one global transaction coordinator to another as the mobile host moves from one mobile cell to another.

d. Disconnection:

Mobile transactions are submitted from a mobile host, and sub transactions are executed at local database servers.

When the mobile host is disconnected, the global transaction is marked as disconnected if the disconnection is known and planned. The execution of the global transaction is still carried out at the local database servers. On the other hand, if the disconnection is unplanned, the global transaction is suspended. The global transaction is resumed when the mobile host reconnects to the mobile support station.

e. Distributed Execution:

Mobile transactions are submitted from mobile hosts, and the entire transactions are distributed among local database servers through the support of mobile support stations. The mobile hosts do not take part in the execution processes.

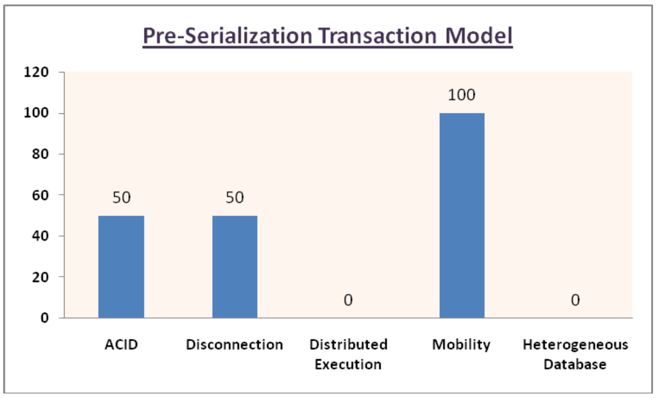

Figure 6: Bar Chart of Pre-Serialization Transaction Model

Since PSTM does not support atomicity and durability, so the performance of this model in transaction properties is 50%. PSTM only supports the planned disconnection. Mobility is fully supported by PSTM.

G. Moflex transaction model (MTM):

a. Description:

The Moflex transaction model [11] [12] is an extension of the Flex transaction model to support mobile transactions. The Moflex model is built on top of multi-database systems and based on the concepts of split-join transactions. The main characteristics of a Moflex transaction are:

a) Each Moflex transaction *T* is accompanied by a set of success and failure transaction dependency rules, hand-over control rules, and acceptable goal states. Dependent factors that include the execution time, cost and execution location of transactions are also specified in the definition of the Moflex transaction. Furthermore, joining rules are provided to support the join of the split sub-transactions (sub-transactions are split when the mobile host moves from one mobile cell to another).

b. Transaction Properties:

The mobile transaction managers make use of the two-phase commit protocol to coordinate the commitment of the Moflex transaction. The Moflex transaction commits when its sub-transactions that are managed by MTM have reached one of the acceptable goal states, otherwise it is aborted. A compensable sub-transaction is locally committed, and the results are made visible to other transactions. For non compensable sub-transactions, the last mobile transaction manger, which corresponds to the end location of the mobile host, plays the role as the committing coordinator.

c. Mobility:

The mobility of transactions is handled by splitting the sub-transaction, which is executed on the local database at the current mobile cell, as the mobile host moves from one mobile support station to another (with the support of the mobile transaction manager). Hand-over control rules must be specified for each sub-transaction. If a sub-transaction is compensable and location independent, it will be split into two transactions; one will continue and commit at the current local database, the second will be resumed at the new location. If the sub-transaction is location dependent, at the new location, the sub-transaction must be restarted. If a sub-transaction is non compensable, the sub-transaction is either restarted as a new one in the mobile cell if it is location dependent, or continued if it does not depend on the location of the mobile host.

d. Disconnection

Moflex transaction model does not support disconnected transaction processing. The Moflex transaction model requires network connectivity between the mobile host and the mobile support stations during the execution process.

e. Distributed execution

The execution of a Moflex transaction is transferred to local database systems at fixed hosts to be carried out there. Moflex transaction model provides a framework to specify the execution of transactions in mobile environments. The main drawback of the Molex transaction model is that the specification of mobile transactions must be fully specified in advance, therefore, the Moflex transaction model may not have the capacity to deal with un-expected or un-planned situations.

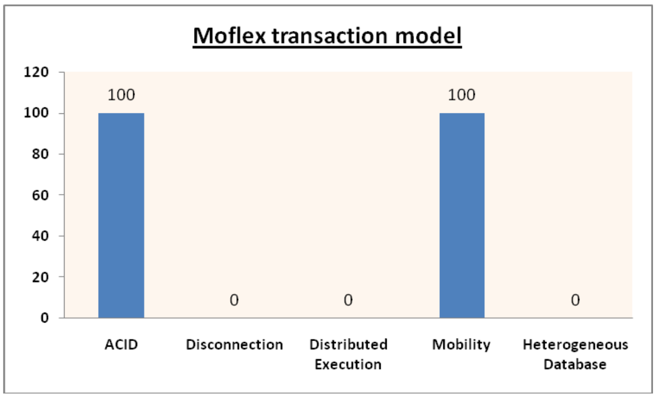

Figure 7: Bar Chart of Moflex Transaction Model

The above figure demonstrates that MTM supports ACID and mobility.

Performance evaluation of some selected existing mobile transaction models is summarized in Table 1 below.

Table 1: Summary of Performance evaluation of some selected existing mobile transaction models

Model Name	Atomicity	Consistency	Isolation	Durability	Disconnection	Distributed Execution	Mobility	Heterogeneity
Kangaroo transaction model	Yes	No	No	No	No	Yes	Yes Partially	No
Reporting and Co-transaction model	Yes	Yes	Yes	Yes	No	Yes	No	No
Pro-motion transaction model	No	No	No	Yes	Yes	No	No	No
Two-Tier(Base - Tentative) transaction model	Yes	Yes	No	Yes	Yes	Yes	No	No
Weak-Strict (Clustering) transaction model	No	Yes	No	No	Yes	Yes	Yes Partially	No
Pre-serialization transaction model	Yes	No	No	Yes	Planned-Yes Unplanned-No	No	Yes	No
Moflex transaction model	Yes	Yes	Yes	Yes	No	No	Yes	No

III. RESULT AND CONCLUSION

This evaluation based on five requirements (ACID properties, mobility, disconnected and distributed execution and heterogeneity) to measure the performance of existing mobile transaction model (Kangaroo transaction model, Report and Co-transaction model, Two-tier transaction model, Pro-motion transaction model, Weak -Strict transactions model, Pre-serialization transaction model and Moflex transaction model) shows that the some of the selected existing mobile transaction models support numerous issues like mobility, disconnection, distributed execution, transaction properties. These results finally tell us where we are lacking in design the mobile transaction model. These will help us improve and control the transaction processing system and will improve the quality of service of mobile devices also. These performance evaluation indications are treated as checkpoints in the future. All the models which have been evaluated, have not taken into account the feature of heterogeneous database, so the researchers can do work on this issue by incorporating existing models or by proposing a new model.

IV. REFERENCES

[1]. P. K. Chrysanthis, “Transaction Processing in Mobile Computing Environment”, IEEE Workshop on Advances in Parallel and Distributed Systems, pp77-83, 1993.

[2]. R. Hirsch, A. Coratella, M. Felder and E. Rodríguez:” A Framework for Analyzing Mobile Transaction Models”, Journal of Database Management, 12(3): pp 36-47, 2001.

[3]. S. K. Madria, M. K. Mohania, S. S. Bhowmick and B. K. Bhargava,”Mobile data and transaction management”, Information Sciences, pp 322-338, 2000.

[4]. Lars Frank, “Atomicity Implementation in Mobile Computing”, in Proceeding of 10th International Workshop on Database and Expert System Application , pp: 105, sep 1999.

[5]. Lisa Clark, ¨Omer Erdem Demir, “Transaction Management in Mobile Distributed Databases” 2003.

[6]. M. Satyanarayanan, “Fundamental Challenges in Mobile Computing”, in Proceedings of fifteenth annual ACM symposium on Principles of Distributed Computing, pp: 1-7, 1996.

[7]. Ayse Yasemin Seydim, “An Overview of Transaction Models in Mobile Environments”, 1997.

[8]. T. Tmiclinski and B. R. Badrinath,” Data management for mobile computing” Sigaiod Rfcord, 22(1):34-39, March 1993.

[9]. Jim Gray, Andreas Reuter, “Transaction Processing: Concepts and Techniques”, Morgan Kaufmann Publishers, 1993.

[10]. Sharmila John Francis , Elijah Blessing Rajsingh, “Performance Analysis of Clustering Protocols in Mobile Ad hoc Networks”, Journal of Computer Science 4(3),pp : 192-204, 2003.

[11]. R. Alonso and H. F. Korth. “Database System Issues in Nomadic Computing”, in Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 388 – 392, 1993.

[12]. W. Booth, G. G. Colomb and J. M. Williams: “Transaction Processing Models in Wireless Network”The Craft of Research, University Of Chicago Press, 1995.