COMPARISON OF AGENT APPLICATIONS IN DATABASE TRANSACTION

Suhaila Mohd Zain
Department of Information Technology and Communication, Kuala Terengganu Polytechnic
Email: suhaila_mdzain@pkkt.edu.my

ABSTRACT

Currently demands for mobile applications have increased to cater the needs of fast and vast data accesses. Therefore, emerging technologies of mobile database transaction have given attention of many researchers to find solutions for new model and techniques for problems in the area of database transaction management. Problems identified in transaction management of mobile database came from the features of the network structure in mobile topology, database operation nature and also limited resources. The purpose of this presentation is to provide an overview of what are the overall architecture lies within, these includes the network structure and the components involved. This presentation also discussed the type of database itself, the transaction and query processing and the techniques to cater main problems in database transaction which is disconnection problem. Finally several examples of agent models and the implementations by previous research also analysed to get the whole idea of this research.

KEYWORD: Database transaction, agent, mobile, disconnection

1. INTRODUCTION

Many early database applications maintained records for large organizations such as corporations, universities, hospitals and banks. The success of database systems in traditional applications encouraged the developers of other type applications to attempt to use them. Currently useful information not only can be extracted from these databases in static, instead, various of device can now be used when a user want to access information in mobile while the information changes according to the user’s movement anywhere and at any time. The mobility has affected the way data is disseminated, query is processed and transaction is managed. The current phenomena that we are in now are called mobile environment.

1.1 Concepts

Mobile computing involves mobility of hardware, data and software in computer applications. It can be viewed as a specialized class of distributed systems consisted of either client-server model or peer-to-peer model. The connections between hosts dynamically changed. It involves computations and data transfer over distributed and mobile nodes, not just voice communication such in mobile communications and also it not necessarily point-to-point or person-to-person as in telecommunications. It involves mobility, but not necessarily with wireless computing, that is wireless does not imply mobility (Ahmad, 2005). To simplify, we can say wireless computing aids mobile computing. There are issues such as mobile computing without wireless networks? Nomadic computing - often means user mobility without device mobility. Three mobility paradigm viewed from user side can be identified as static, nomadic and continuous mobility. Nomadic according to S. Uskela (Uskela, 2002), “the ability to move easily from place to place and retain access to information and communication services while moving, at intermediate stop and at the destination”, but the nomadic currently refer to portability. Alonso and Korth (Alonso & Korth, 1993) stated that nomadic computing refers to mobile computers in integrated part of distributed computing environment. Meanwhile continuous mobility can be defined where users are always reachable and network provides reasonable communication capabilities for moving users (Uskela, 2002). Further explained that mobile motion of objects can be two approaches, user movement (personal mobility) or movement of user’s device (device mobility).
2. LITERATURE REVIEW

2.1 Components Involved in Mobile Computing

According to previous research (Ahmad, 2005; Dunham & Helal, 1995; Le Gruenwald, Banik, & Lau, 2007; L. Gruenwald & Banik, 2001; Kumar Madria, Mohania, Bhowmick, & Bhargava, 2002; Serrano-Alvarado, Roncancio, & Adiba, 2004), mobile computing environment basically includes:

i. A fixed network which is a distributed system
ii. Fixed Host (FH) typically is a computer in high speed fixed network but has no connection with MH.
iii. Mobile Unit (MU) or Mobile Host (MH) or sometime called Mobile Terminal (MT), is a mobile computer connected to fixed network via a wireless link. Some would prefer Mobile Station (MS) to refer to same thing. MH in some respect may also refer to sites.
iv. Base Station (BS) or Mobile Support Stations (MSS), using wireless interface to connect to a MU.
v. Database Server (DBS) has database processing capability.
vi. Cell, where each BS/MSS is responsible for all of the MHs in a given logical area. A cell can be WLAN, ad hoc network or a cell phone network.

Mobile and fixed hosts can be either clients or servers and MHs can be of different natures, from PDAs to personal computers. An MU may change its location and network connections while computations are being processed. In cellular wireless networks MTs only communicate with the BS but not with each other.

Here, no specific assumption are made about the database model (relational, object) and the centralized or distributed nature of the database management system (DBMS) (Serrano-Alvarado, et al., 2004). The same applies to models in distributed system which has been defined in previous section (peer-to-peer / client-server). To understand the movement of MU, handoff procedure is used as a process of entering a new cell, it can be called hand-off or hand-over (Serrano-Alvarado, et al., 2004). Information of current MH state should be transferred to new BS while crossing the boundaries of a cell (Kumar Madria, et al., 2002).

2.2 Architecture for Mobile Computing System

Basically mobile computing architectures can be categorized into two types (L. Gruenwald & Banik, 2001).

a) General Architecture: consists of fixed BSs/MSSs that maintain all MHs in its cell. Control handover done by new MSS when an MH moves to a new cell. This architecture concerned the mobility of users as MH. Figure 1 shows a topology of mobile computing from prior research paper.

b) Mobile Ad-Hoc Network (MANET) Architecture: consists of MHs that roam in the network. They are connected with wireless network that frequently changed its topology, so there is no fixed infrastructure and fixed MSSs. This architecture has databases which are distributed in heterogeneous community (Brayner & De Aguiar Moraes Filho, 2006). It is dynamically configurable structure suits its name as Ad-hoc. Other defines it as several mobile nodes sharing one or more wireless channels without centralized control or an established infrastructure.

![Figure 1: Design of General architecture used by P. Serrano-Alvarado, et al (Serrano-Alvarado, et al., 2004) in mobile transactions.](image-url)
Dong-Soon, Hae-Kyung, & Ung-Mo, 2001) added that each MH includes several applications and one small DBMS. The function is to retain database consistency of local applications transaction. Replication control tools are then used to maintain replicated data in mobile database environments for data synchronization by multiple MHs.

In another paper, Madria et al. (Kumar Madria, et al., 2002) added that both BS and FH need DBS help to perform transaction management. The DBSs can be set either at BS or FH. They suggest that, most commonly used applications are provided by BS, thus gives advantage to MH which has power and storage constraints.

Serrano et al. (Serrano-Alvarado, et al., 2004) considered that MHs can store and run DBMS modules. The communication capabilities between MHs and BSs are asymmetric. And it was assumed that no direct communication between MHs, so this limits the communication just by BS. Another unique features stated by Khder et al. (Khder, Zin, Zulaiha, & Othman, 2008) that MH can initiate a transaction to disseminate it directly among a set of MH or FH. However, there must be specific protocol to support movement of these MH while accessing data from MH through the nearest BS.

Gruenwald and Banik (L. Gruenwald & Banik, 2001) stated many previous researchers employed the first architecture for mobile database transaction management. However they raise issue in MANET, an architecture which concerns on movement of database stored in MH. The added feature to the architecture is that two MHs of different cell can communicate using intermediate MHs between them. This has made it a dynamic and effective architecture that will enhance the transaction processing resource (Brayner & De Aguiar Moraes Filho, 2006; Eder, Missikoff, Brayner, & Filho; Le Gruenwald, et al., 2007; L. Gruenwald & Banik, 2001).

### 2.3 Database and Data Management in Mobile Computing

There are heterogeneous database systems available in mobile database computing, examples include, object oriented database, relational database, XML type, or others (Eder, et al.). Regarding this scenario, two types of database system reside in the mobile environment, one which located in fixed host and another located in mobile host. The latter represent mobile database (Eder, et al.).

A database which stores locations of users will often changes because of user’s location varies at times. According to Lam et al. (Lam, Kuo, Tsang, & Law, 2000), the database is structured to local databases and distributed at different BSs. It consists of temporal and non-temporal data objects. Different between both is that temporal data objects are used to record object status in the external environment. Timestamp is associated to each temporal data object to trace the data object life. The timestamp is given when a transaction is initiated and set upon it if the transaction successfully commit before its deadline (Lam, et al., 2000).

There are two operation regarding data broadcast in the environment, push-based and pull-based. Distributing information to clients without requests is a push-based whereas clients request data from server is pull-based operation (Barbara, 1999). The fact is then expanded by Rahbar et al (Rahbar, Mohsenzadeh, & Rahmani, 2009), stated there are three communication methods in MANET, broadcast, data query and peer to peer messages. However, not all types have been used in an application altogether. The broadcast is the most practical approach for conserving energy in which the other MUs just need to listen and forward relevant message or query.

Management of data in mobile and distributed environment differs in such a way where the distributed application focused on “location transparency” whereby in mobile application focused on “location awareness” (Dunham & Helal, 1995).

According to (Eder, et al.; Imieinski & Badrinath, 1993; Kumar Madria, et al., 2002) data management in mobile computing can be described as global and local data management. Global data management relates with network aspect such as location and addressing to find routes and send packets, replication such in copying data as local and broadcasting to discovers and exchange messages. On the other hand, local data management refers to the end user point of view, for example energy utilization from data access operation, disconnection management possibly from handoff and query processing regarding atomic transaction.

Characteristic of mobile database in mobile computing environment have been analyse by many researchers (Ahmad, 2005; Barbara, 1999; Imieinski & Badrinath, 1993; Kumar Madria, et al., 2002; Walborn & Chrysanthis, 1997) from various aspect, such as it impact to data management, it features to user choices and it constraints to devices. Below are listing of the characteristics:

i. Asymmetry in the communication
ii. Energy constraint /Power limitations
iii. Size of screen.
iv. Communication in a cell using broadcast
v. Limited resources, such as storage capacity
vi. Varies locations at times (dynamic topologies)
vii. Scalability or coverage
viii. Limited security (vulnerable to intruders)
ix. Frequent disconnections (and predictable disconnections)
x. Greater mobility implies weaker network connection

xi. Bandwidth availability (effect quality of connections, more nodes will reduce network capacity).

Kottkamp and Zukunft (Kottkamp & Zukunft, 1998) consider that querying processing are basic task in mobile database systems. In their paper they discuss an approach to optimize query regarding fast changing location in mobile database systems. A query will depend to the state of the MU (connected or disconnected) thus executed differently.

Some query in database may refer to location dependent data. Gok and Ulusoy (Gök & Ulusoy, 2000) stated that moving object changed position continuously as a function of time. Query can also be identified as remote access to data in heterogeneous MDBC. The MDBC need to be integrated because it includes many mobile databases to process query (Eder, et al.). These queries can refer to both mobile nodes and fixed nodes accessing data through network connection characterized by constraints as listed. These nodes made connections to the network to access data either through land-based lines (a modem and telephone lines) or through a wireless medium. User interaction in a mobile database is accomplished through transactions.

2.4 Mobile database in Mobile Ad hoc Network (MANET) architecture

Specifically, mobility covers movement of object (users or devices) between geographical location, different network, changes in communication devices and different application. In this research, physical mobility of devices and logical mobility of database transactions is considered. However, whatever the aspect of mobility is, continuous connectivity of wireless access must be supported especially in ad hoc networks.

There existed two different ways of configuring a mobile network, infrastructure based and ad hoc based. The first used fixed network access points (base stations) such in Wireless LAN (IEEE 802.11) and global wireless networks (GSM, GPRS, UMTS). While an ad hoc network existed for the duration of the communication only. This refers to the nature of the network being transitory and only coming together for a short time when it is needed without the need for permanent infrastructure (Golding, 2004). Due to the lack of infrastructure, mobile units (MU) act as routers and forward packets to others as depicted in Figure 2. Each MU periodically broadcasts its power level and location information to each other.

![Figure 2: Dynamic topology and the communication in MANET (Technology, 2001)](image)

The infrastructure has dynamic topology, in which the MUs will get in and out from the network without fixed setup. MANET applications have common characteristic such that they have same logical group of data accessed from the network. Another is, it moved randomly, but still within a finite distance for example 802.11b wireless distance is 100 meters. In the mobile computing environment, each MU has its own database system (Jing & Jianhua, 2009). It changes its location while the computation is being processed. Many applications apply this feature for example in tactical networks, emergency services, entertainment, location-aware services, etc.

However the popularity of MANET application comes with restriction in resources. Due to this problem many research discussed on how to manage the resources effectively. Some works have tried to reduce energy consumption by assigning each MU with different modes. The modes identified are, Active, Doze, and Sleep, as follows (Le Gruenwald, et al., 2007; L. Gruenwald & Banik, 2001; Kumar Madria, et al., 2002):

- **Active Mode**: The MU performs its usual activities. Its CPU is working and its communication device can transmit and receive signals.
- **Doze Mode**: The CPU of the MU will be working at a lower rate. It can examine messages from other MUs. The communication device can receive signals. So the MU can be awaken by a message from other MUs.
Sleep Mode: Both the CPU and the communication device of the MU are suspended because of communication lost (Jinghua, Zhiyun, & Minhua, 2010). This is equivalent to a system failure considered in classical transaction processing work.

Data accessibility in MANET is lower compared to fixed networks. In order to overcome this problem data items are replicated at MU, in which the MU may not be the original data host. However this solution is difficult to realize because of poor resources and according to Hara and Madria, it is impossible for MU to replicate all data item in the network (Hara & Madria, 2004). So one possible way is that replication needs to be made in selective way, for example to replicate data to MU that have enough resource or to replicate popular data item.

3. METHODOLOGY AND FINDING

The methodology in this research focuses on the execution of query processing from mobile transaction in ad-hoc network. Types of transaction are also discussed to see the difference in transaction processing. Then followed by agent application which is the finding to overcome transaction problem in the area of research.

3.1 Mobile Transaction and Query Processing

Elmasri and Navathe (Elmasri & Navathe, 2004) define a transaction as a program consists of database operations: read set, such as select statement to retrieves the dataset; and write set, this includes all insert, update and delete statements. The write set includes begin and end transaction statements. The transaction effect is made directly on the database resulting commit or abort (rollback). Figure 3 illustrate the flow of transaction states.

![State transition diagram of a database transaction](image)

In executing transactions many events happened with or without prior notice. In order to associate the situation, status of transaction operations can be identify as one of the following (Kumar Madria, et al., 2002):

- fully connected (normal connection);
- totally disconnected (e.g., not regarded as a failure of MU);
- weak connection (a terminal is connected to the rest of the network via low bandwidth).

In mobile environment, at least one MH is involved to call it a mobile transaction(Serrano-Alvarado, et al., 2004). There are suggestions on where mobile transaction management should be relied to. First idea was to put it as MH’s responsibility, second suggestion said BSs should handle it, and third option suggested that the management moved with the MH which requested the transaction (Khder, et al., 2008).

Many data sets can be executed through multiple BSs in a single mobile transaction. This is due to the MUs movement. The transaction result is atomic in that every operation in the transaction either completes, or does not complete. In other words, transactions are comprised of read/write operations with commit or abort operation at the end (Lim & Hurson, 2001).

Example transaction can be seen in mobile commerce (Tsagatidou & Pitoura, 2001), where a traveler has request information for nearest hotel. He paid a hotel room and made transaction. To succeed in the transaction, the properties of ACID need to be fulfil. By executing all operation of the transaction successfully, the atomicity can be achieved.
In previous researches (Alshorman & Hussak, 2008; Brayner & De Aguiar Moraes Filho, 2006; Eder, et al.; Lim & Hurson, 2001) stated there are 2 kinds of transactions to be executed concurrently in MDBC, these are mobile transactions and local transactions. The difference of both transactions is that mobile transactions do not have one computer site which serves as the transaction management site (Lim & Hurson, 2001).

To ensure the correctness of these transactions, serializability needed to adhere. It means that all concurrent of local and mobile transactions should be executed in schedule which effected as a synchronized execution of the transactions. Or in other words it depicted that the transaction is executed in isolation without interference from other transaction steps.

From multi-database system (MDBS) perspective, Dunham et al. (Dunham, et al., 1997) viewed a mobile transaction not as a global transaction (global transaction are transaction submitted through a global interface which may consist of sub-transactions at multiple local sites (Lim & Hurson, 2001)).

In other research, Chang et al. (Chang et al., 2007) classify two basic transactions as online transaction and offline transaction by referring to owner of transaction. Mobile transaction manager directly interact with online transaction at the DBSs attached to FH. However, in offline transaction the transaction manager will execute and manage the disconnected MH. Problems during transaction operation can be due to frequent aborts that will resulted delays to the user request and then effect the database consistency of facing deadlock (Lim & Hurson, 2002).

The nature of transactions that moves through the network thus makes a difficult task to satisfy ACID properties. ACID properties have been regarded as the indicator to satisfy the successful of data communication in mobile computing. These properties impact to the transaction management is concluded in Figure 4.

![Figure 4: Conclusion of data management and transaction processing. (Elmasri & Navathe, 2004)](image)

3.2 Agent Model in Networking

The use of agent has shown significant success in current applications, especially regarding web services. An agent is autonomous program that can act and react on its’ own. They adapt to the environment in which they resides with capability to achieve series of user designed goals especially in mobile computing (Jing & Jianhua, 2009). In other words, the mobile codes can do it jobs as prescribed by the system (Eder, et al.).

Generally the purpose of using agent was to control and handle transaction management. The usage of agent application in the networks mainly because of it can move freely and can be customized into the environment. Thus further gives efficiency and flexibility to the transaction management (Jing & Jianhua, 2009).

Characteristics of an agent model according to (Pitoura & Bhargava, 1995) are listed below:

i. There are no global transaction manager since agent based computation is decentralized.

ii. Agent is dynamic in which can be modified at runtime, it decomposition depends on previous actions.
iii. The interface consists of primitive method from each database system involved

4. ANALYSIS

This section is first to explain an overview of situation in mobile database transaction and then the management of transaction to established ACID properties which is recovery and concurrency control management. Then the finding is continued by application of agent in applying the control management methods.

4.1 Challenges in Mobile Database Transaction

Significant distinction between mobile transaction management and distributed transaction management is transaction movement through the network. This is due to its behavior to access data that may change location or change according to changes in one of the component in the architecture. Another distinct feature can be seen in the transaction model because a mobile transaction is only identified by the collection of cells it passes through. The execution of the mobile transaction is thus not fully coordinated by the system.

Transaction control can move along with MH or remain at originating cell when it hops to a new cell. Messages must be sent to new BS if transaction control remains at the originating site because MU needs previous information. To handle messages overload, the transaction control must moves with the MU (Dunham, et al., 1997).

Mobile transaction management system includes problem related to replication control, mobility control and failure handling (Jianting & Wanlei, 1998). Ensuring reliability of shared data becomes difficult because of complexities in mobile computing structures added with limitations and problems in communication channels. Possible problems during handoff may increase the rate of transaction failures, such as changing signal strength, which then disconnect MH from the network. Partitioning of the network due to MU failures can complicates process to update and to find routes.

Disconnection can happen when a connection link of a mobile host is broken accidently or intentionally. However, it must be handled as “normal” situations because MHs may disconnect to save battery consumption (Serrano-Alvarado, et al., 2004).

A MH should be set to automatic operation even in disconnection mode. Special action must be taken to cater predicted disconnection over active transaction, several identified steps are (Kumar Madria, et al., 2002):

- Migrating transaction process to FH if the transaction needed no further user interaction.
- Data needed in transaction can be downloaded in advance before disconnection, so the transaction can continue executing locally on the MH after disconnection.
- Transferring log records from MH to FH, a precaution for instability of storage.
- The MH can announce itself upon predicted disconnection by disabling any connection, this to give alert for distributed protocol to handle the disconnection.

To ensure proper support for mobile transactions (example; transactions initiated on a MH), local autonomy must be given to allow transactions to be processed and committed on the MH regardless of constraint such in disconnection. One suggestion in (Kumar Madria, et al., 2002; Sagayeshi, Shiri, & Faraahi, 2009) for prevention of transaction abort is by using cache in MUs, this further handle frequent disconnection of relocation and low bandwidth problem. In doing it the availability of wireless connection can be optimized. Caching mechanisms are characterized by its granularity, consistency strategy and replacement policy. Lee et al. (Lee, Hwang, & Yu, 1999) rise issues of maintaining transactional cache consistency for mobile computing because of frequent disconnection. Narrow bandwidth would be clogged if massive numbers of mobile clients attempted to query a server to validate cache data. They discuss broadcast facility to support transaction in mobile database system.

Other mechanism to manage disconnected operation is by using hoarding, a technique to preload data on mobile device to allow continued operation, then to serialize log update and integrate the database upon reconnection (Satyanarayanan, 1996). Another paper introduced transaction processing model as Replication Method, which created algorithms that would release strict consistency in replicas. It is an alternative to handle site failures in a mobile environment (Jianting & Wanlei, 1998). Techniques in caching schemes and data replication has been rebuilt and re-evaluated to adapt new model and architectures which synchronously emerge with technology advancement, thus create ideas for problem solutions (Dunham & Helal, 1995).

4.2 Recovery and Concurrency Control Management
Replication scheme and caching scheme are data duplication techniques applied in mobile transactions. According to (Lim & Hurson, 2002), both scheme differ in scope of the data access. In the first scheme, other outside system can access the replicated data, while cached data only possible for the system in which the data reside.

Abdul Mehdi et al (Abdul-Mehdi, Bin Mamat, Ibrahim, & Diris, 2006) proposed Multi-Checkout Timestamp Order (MTCO), the idea is to allow replicated data of MU to access and update with low cost. By applying peer to peer and replication concept MCTO in planned disconnection mode, nodes have to declare their intention and take the object for writing. This is done by obtaining lock on the data item before disconnection and the use of timestamp to serialize the transactions after the mobile node reconnect to the network.

The approach of using timestamp has similarities with framework by Khder et al. (Khder, et al., 2008), a method to manage data called Fixed End Timeout Technique (FETOT). The method will managed planned disconnection in mobile database over fixed network with weighted data distribution. In their paper they discussed about connection time accessed by user, because the time will determine transaction management. Its function was to reduce delay in total commit time for time-out and then to allocate data to MH and FH as requested using timestamp order.

Walborn and Chrysantis proposed PRO-MOTION (Walborn & Chrysanthis, 1997), a transaction processing model to support disconnected database operations between Database server and MU. The technique is to use agent named Compact, which derived to process transaction request executing at MU. The basic idea is to permit global constraints to be transformed into local constraint in achieving database consistency. Therefore, the model can support dynamic replication as well as caching. However the interaction of Compact agent and compact manager is processed by mobile support station (MSS). So the architecture relies on MSS to handle the transaction.

4.3 Agent Application in Transaction Processing

Brayner and Filho (Eder, et al.) proposed an agent based model in MANET called Accessing Mobile Database (AMBD). The model proposed to support heterogeneous mobile database and query processing concerning concurrency control and recovery for mobile and distributed transaction. They assume two-phase commit (2PC) protocol is enforced to ensure atomicity in global transactions since all existing database implement the protocol. Mobile agent named Wrapper created Runner agent to manage global transaction from user, then it will create another agent called Broker as a backup for coordinating 2PC process. The interaction of agents involved in the architecture is depicted in Figure 5.

![Abstract model of AMDB](image)

The model replaced central coordinator role by creating sequence of agents therefore supports physical and logical mobility. However too many agents in the architecture will complicate the task and consume more energy.

Pitoura and Bhargava (Pitoura & Bhargava, 1995) in their paper, proposed and agent application with timestamp approach. It works with each application agent receives a timestamp defined to be a combination of the clock value and the user’s id. This timestamp corresponds to a global serialization order, in which each application agent serializes all conflicting methods on its local data based on the timestamp order. An operation on its local data issued by another agent is executed only after ensuring that the two agents are allowed to “meet” at a break or relocation point.

MDB model by Younas et al (Younas, Chao, & Anane, 2003) considered m-commerce transaction with agent support. Management of mobile transaction is done by three different agents; they are planning, execution and recovery agents. These agents must kept record of a mobile transaction, and gather necessary information.
They use timeout techniques to deal with disconnection, in which agents are free to act if the expected duration is already expired. However mutual agreement needs to be specified before implementing the transaction.

Jing, L. and W. Jianhua (Jing & Jianhua, 2009) proposed an architecture based on agent to improve processing efficiency of mobile database system (Figure 6). The model was only applicable on client server architecture. The major functionalities are done by local interface agent who creates four other agents in mobile agent layer. However their work didn’t specify any action to handle disconnection and ways to cater limited communication bandwidth which are part of their goal in proposing the new architecture.

![Figure 6: Mobile database architecture based on agent.](image)

In another research by Sagayeshi et al. (Sagayeshi, et al., 2009) they focused on using dynamic agent to handle cache in MU to prevent transaction abort for disconnection problem. A cache is maintained to keep up with data so that the transaction is not lost due to connection failure. Every agent created by MSS will have an id to indicate its boundary and it associated MUs. It works by predicting information to be sent to MU for cache operation. However, MSS does the prediction by using Record number between current MU and other MUs with weighted tuples in fuzzy database.

Ongtang et al. (Ongtang, Hurson, Yu, & Potok, 2007) proposed AT3M, an agent based transactions management for mobile multidatabase environment. In their paper, the agent algorithm allows global sub-transactions to process in parallel and support user mobility while in disconnection. However the scheme is in client server structure.

To conclude, constraints in mobile database transaction comes with the network structure, mobility features, resource limitation, types of database and various techniques to manage transaction failures.

5. CONCLUSION

This research has identified and discussed two types the architecture and components involved in mobile computing. Data management and the features of mobile database are also has been viewed from various aspect. Further discussed are mobile database in ad hoc network architecture. Some problem regarding with restriction on mobile devices resources were also highlighted.

Then the research went further with methods in the mobile transaction and query processing. Challenges in mobile database transaction has been analysed by focusing on disconnection of the transaction. Several agent models in the application have been reviewed and look in brief for further understanding. Each model has a unique and different approach to cater many types of problems depends on the environment and scope. These include the type of database, database transaction, techniques to solve problem and methods of disconnection.

REFERENCE


BIBLIOGRAFI

Berasal dari Kuala Terengganu
Mempunyai kelayakan Sarjana Muda Teknologi Maklumat UUM (2001) dan Sarjana Sains Komputer, La Trobe, Australia (2011)