Transaction Processing In Mobile Computing

Articles

- **Mobile Computing and Databases-A survey**
  Daniel Barbaral
  IEEE Transactions on Knowledge and Data Engineering,Vol 11.

- **Correctness Criteria for Multilevel Secure Transactions**
  Kenneth P.Smith, Barbara T. Blaustein, Sushil Jajodia
  IEEE Transactions on Knowledge and Data Engineering,Vol 8.

- **Agent-based transaction processing**
  Little, H.; Esterline, A.
  Southeastcon 2000. Proceedings of the IEEE , 2000

Outline

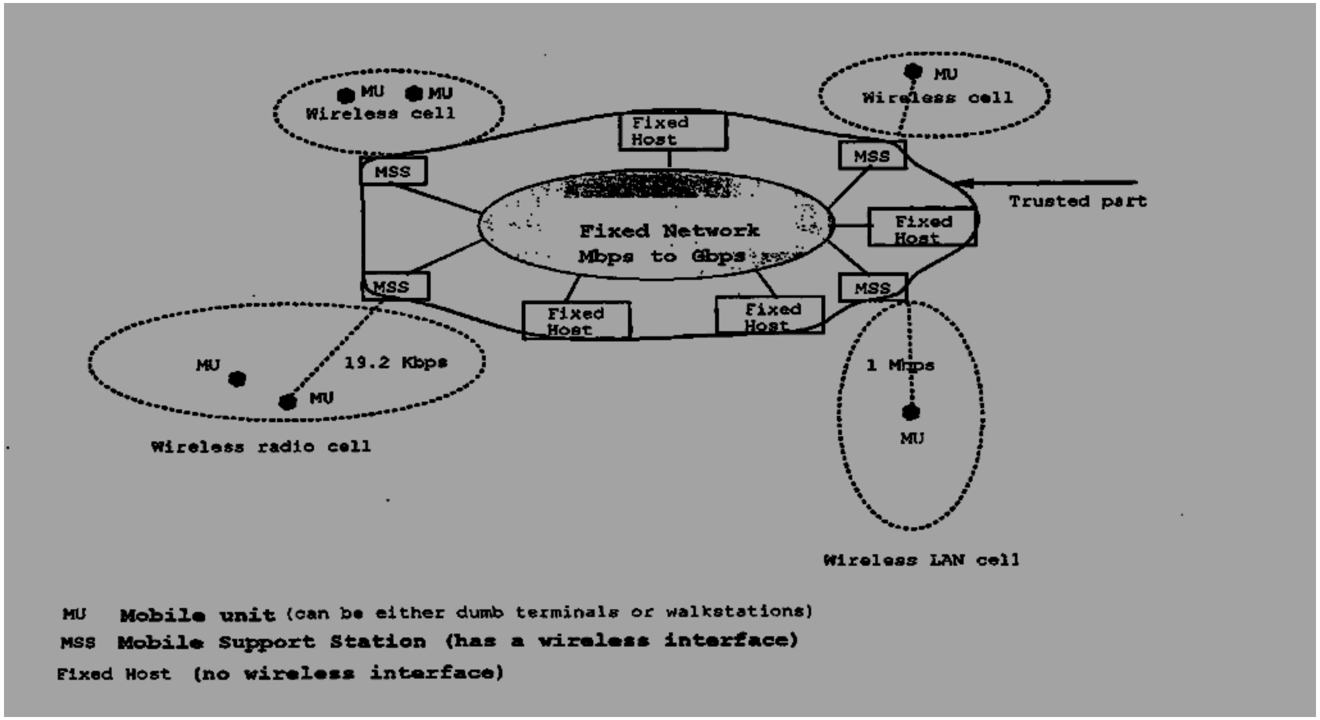
- Transaction management in Mobile Computing
  - General Architecture
  - Unique features
  - Transaction Models
- Model for Multilevel Transactions
  - Background
  - Architecture
  - Methodology
  - Protocols & Interactions
- Transaction Processing using Agents
  - Background
  - Standard Properties
  - Advancements

Transaction management in Mobile Computing

- Why need Mobile Computing:
  - Appearance of powerful portable computers
  - Development of fast reliable network

Architecture of General Mobile Environment

- Two Entities
  - Mobile units
  - Fixed Host

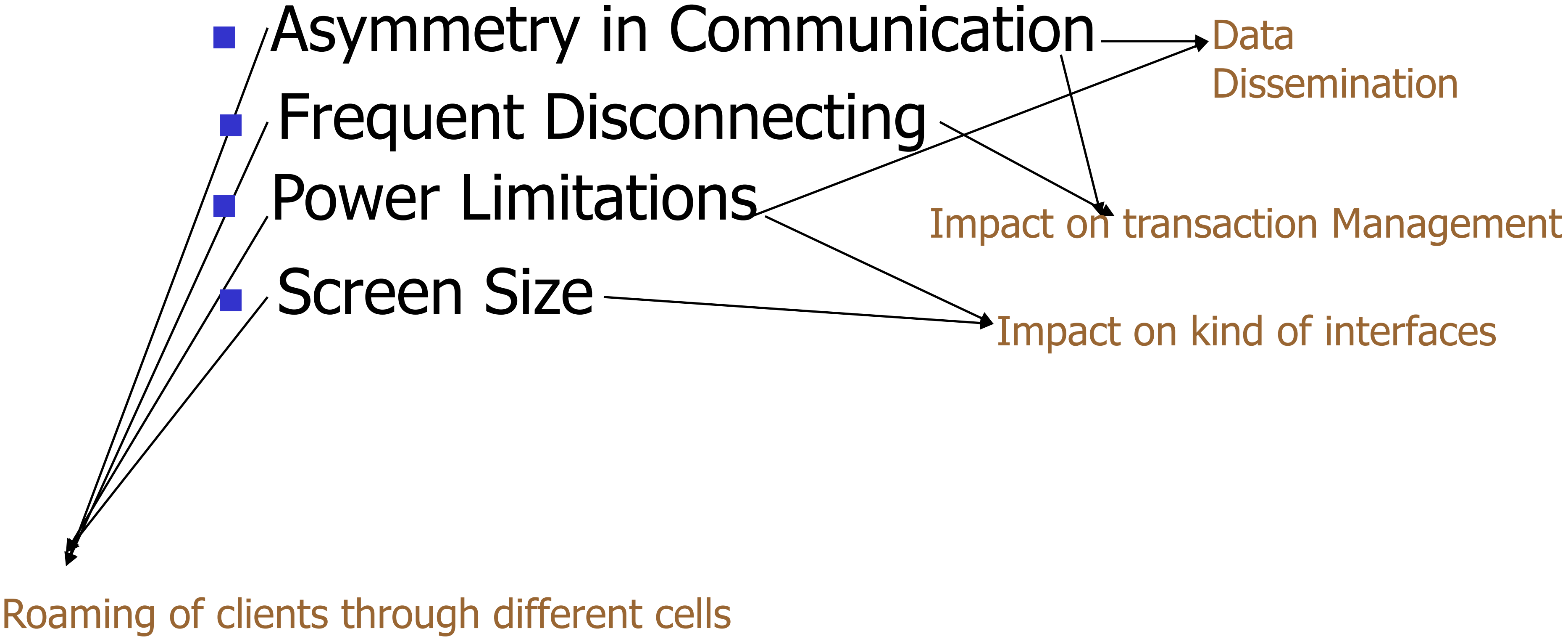


MU Mobile unit (can be either dumb terminals or walkstations)

MSS Mobile Support Station (has a wireless interface)

Fixed Host (no wireless interface)

Unique Features



- Asymmetry in Communication
- Frequent Disconnecting
- Power Limitations
- Screen Size

Data Dissemination

Impact on transaction Management

Impact on kind of interfaces

Roaming of clients through different cells

•Location Dependent Queries Issue

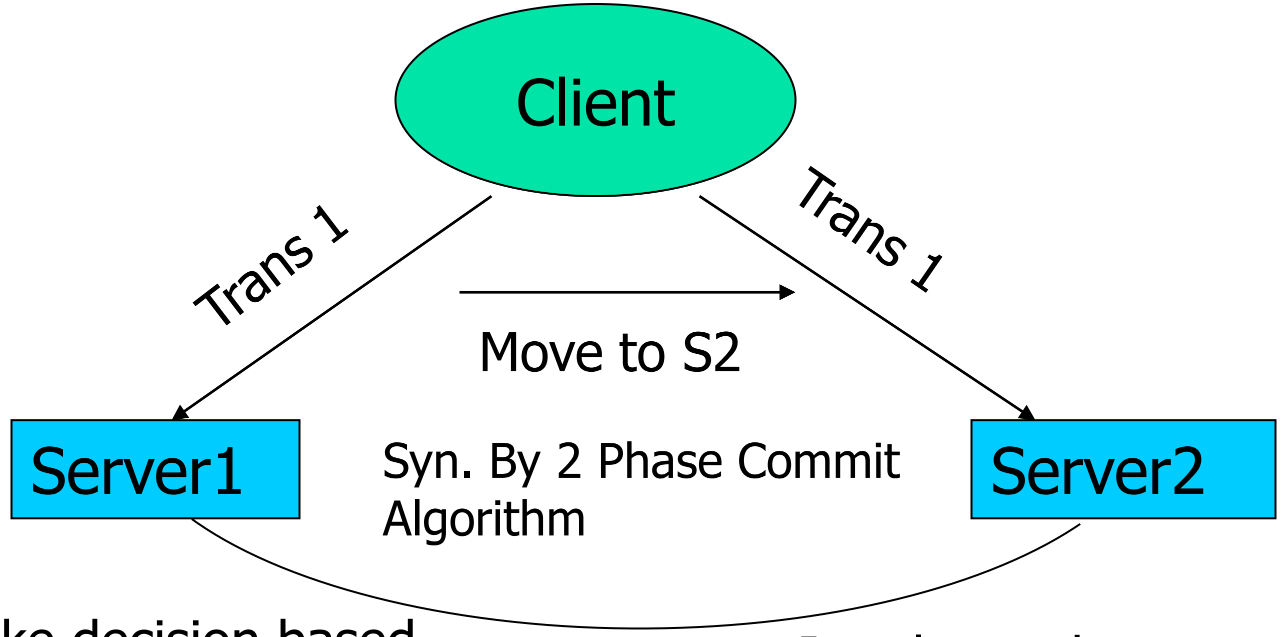
Transaction Models

- Escrow Method
- Walborn & Chrysanthis Generilazation
- Demarcation Protocol
- Two Tier Replication Algorithm
- Certification reports
- Isolation only transactions

Escrow Method

- Divides the total number of instances of an item among no. of sites in system.
- A transaction can only successfully complete at a site if no. of instances it requires does not exceed available instances at that site.

Demonstration



Syn. By 2 Phase Commit Algorithm

It will take decision based on available instances

It only need next operation to be performed

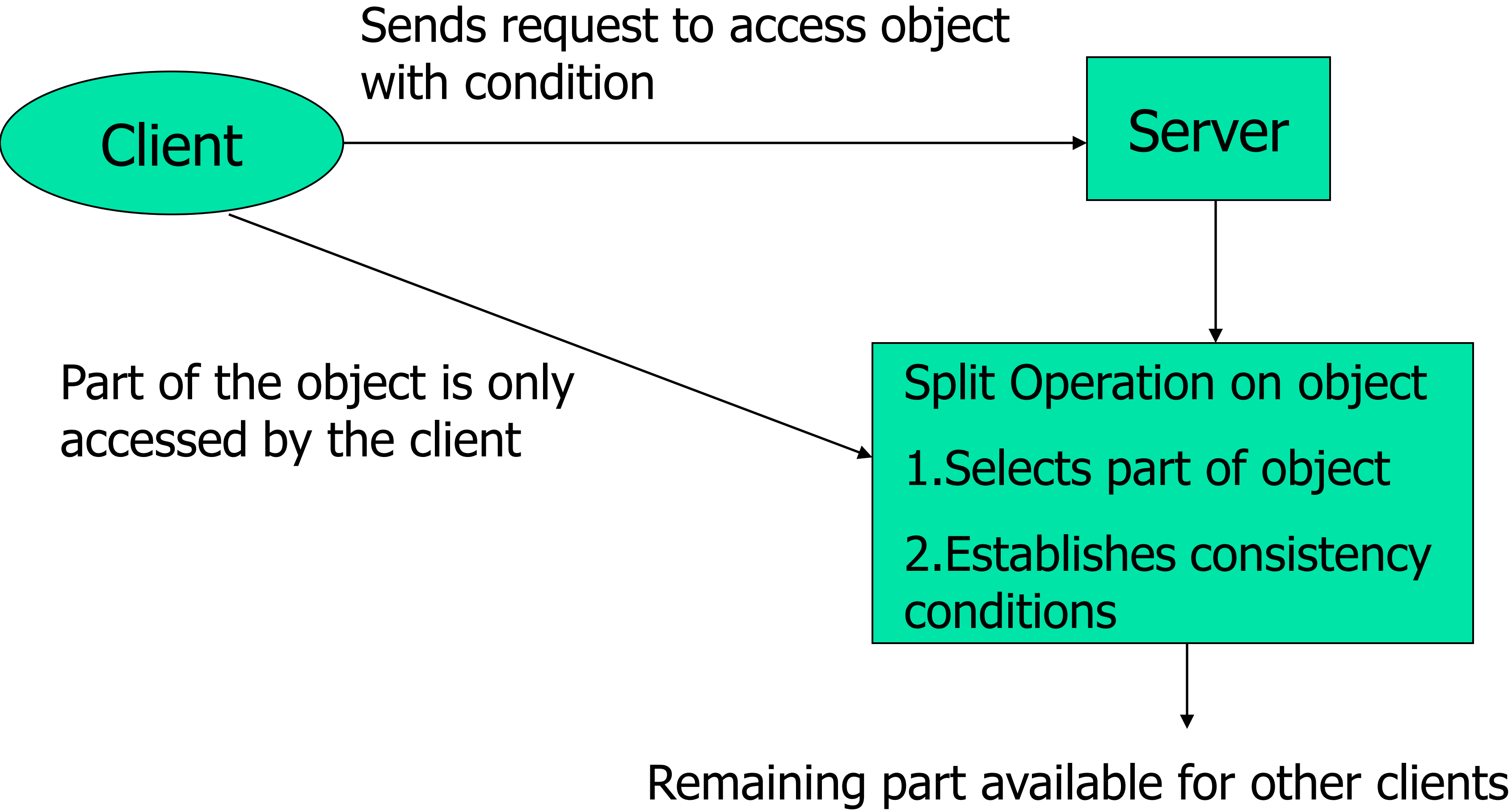
Walborn & Chrysathis Generilazation

- Splits large and complex objects into consistent sets of smaller fragments
- Cache a set on Mobile client

Advantage:

Concurrent Operations on a set of Mobile Clients

Demarcation Protocol



Sends request to access object with condition

Client

Server

Part of the object is only accessed by the client

Split Operation on object

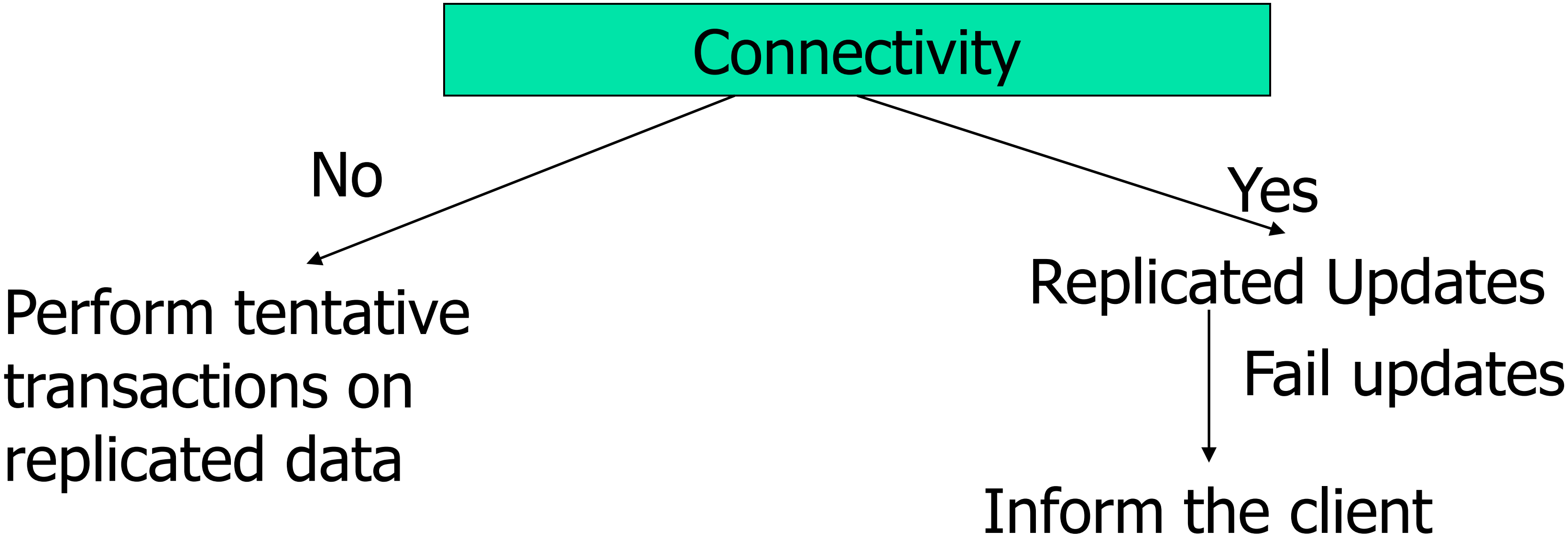
1.Selects part of object

2.Establishes consistency conditions

Remaining part available for other clients

Two Tier Replication Algorithm

- Tentative transactions at mobile clients on replicated data while they are disconnected



Connectivity

No

Perform tentative transactions on replicated data

Yes

Replicated Updates

Fail updates

Inform the client

Two Tier Replication Algorithm (cont'd)

- Advantage:

  Standard way of propagating updates to replicas.

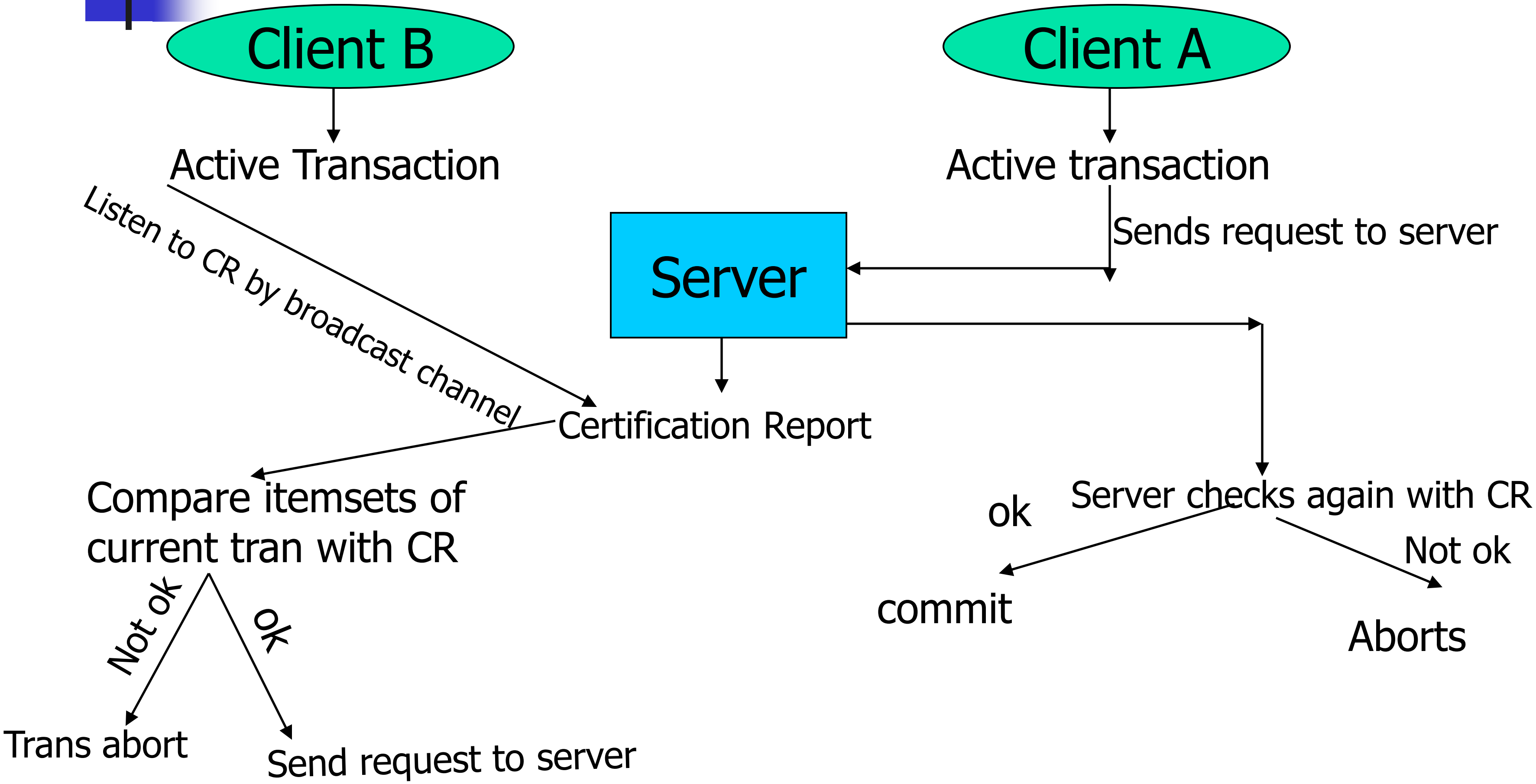
- Disadvantage:

  May result in unacceptable no. of failed transactions

Certification Reports

- Provides a technique with which broadcast channel can be used to help mobile clients to do some verification for a transaction running by them and need to be aborted.

Algorithm



Client B

Active Transaction

Listen to CR by broadcast channel

Client A

Active transaction

Sends request to server

Server

Certification Report

Compare itemsets of current tran with CR

Not ok

Trans abort

ok

Send request to server

Server checks again with CR

ok

commit

Not ok

Aborts

Certification Reports (cont’d)

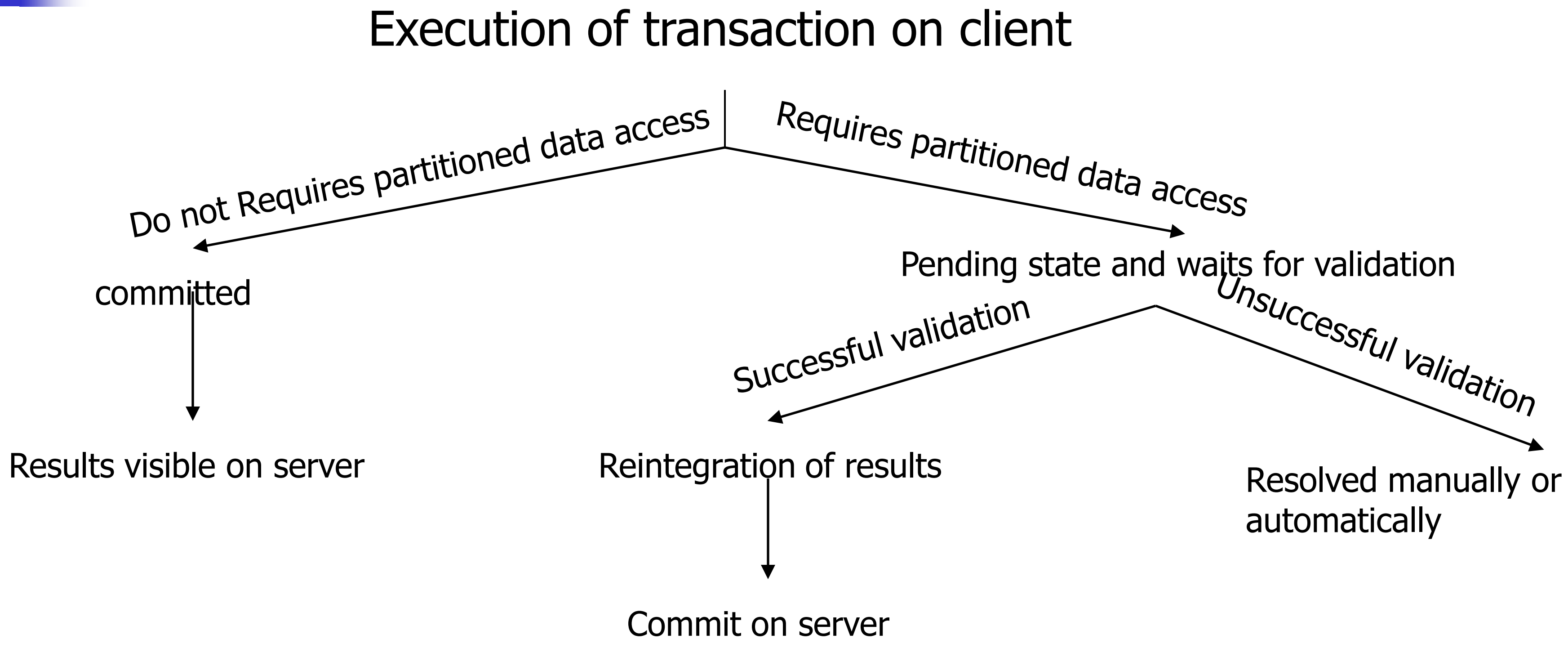
- Advantage
  - Most of the work of verification by clients
  - Early abortion of false transaction

Isolation Only Transactions(IOT)

- Strong Consistency can be guaranteed

- Without guarantying other transaction properties such as atomicity and durability.

Isolation Only Transactions(IOT)-cont’d



Execution of transaction on client

Do not Requires partitioned data access

Requires partitioned data access

committed

Results visible on server

Pending state and waits for validation

Successful validation

Unsuccessful validation

Reintegration of results

Commit on server

Resolved manually or automatically

Model for Multilevel Transactions

- What is Multilevel Security Level:

•Users could only access data on which they have security rights

•No “writing down”

In same session,reading from one level and writing to a lower level.

Current Multilevel Secure DBMS

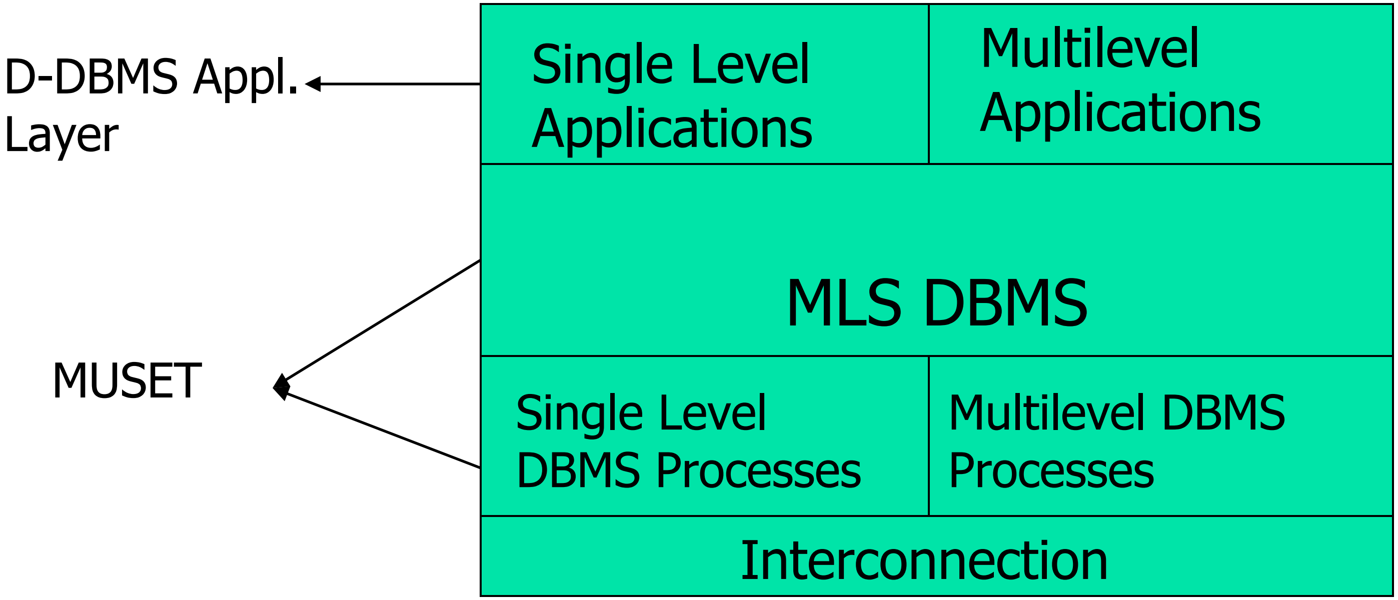
- They implement multi security level.
- No writing down.

Solution :

MUSET (Multi level Secure Transactions) Project

MUSET

- Layered Architecture of MUSET:



D-DBMS Appl. Layer

MUSET

Single Level Applications

Multilevel Applications

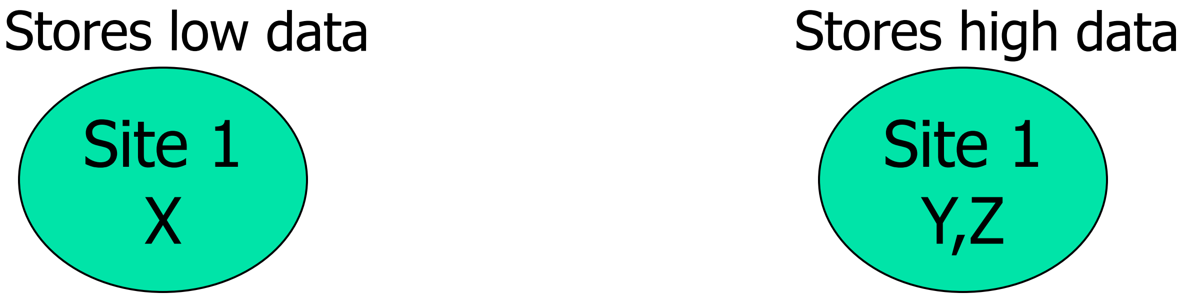
MLS DBMS

Single Level DBMS Processes

Multilevel DBMS Processes

Interconnection

Operation



Stores low data

Site 1
X

Stores high data

Site 1
Y,Z

Let site 2 issue following ML site transparent transaction

Y:=1

X:=X+2

Z:=X+Y

MUSET Analysis

```
Y:=1
X:=X+2
Z:=X+Y
```

High :W(Y)

Low : R(X)

Low: W(X)

High: R(X)

High: R(Y)

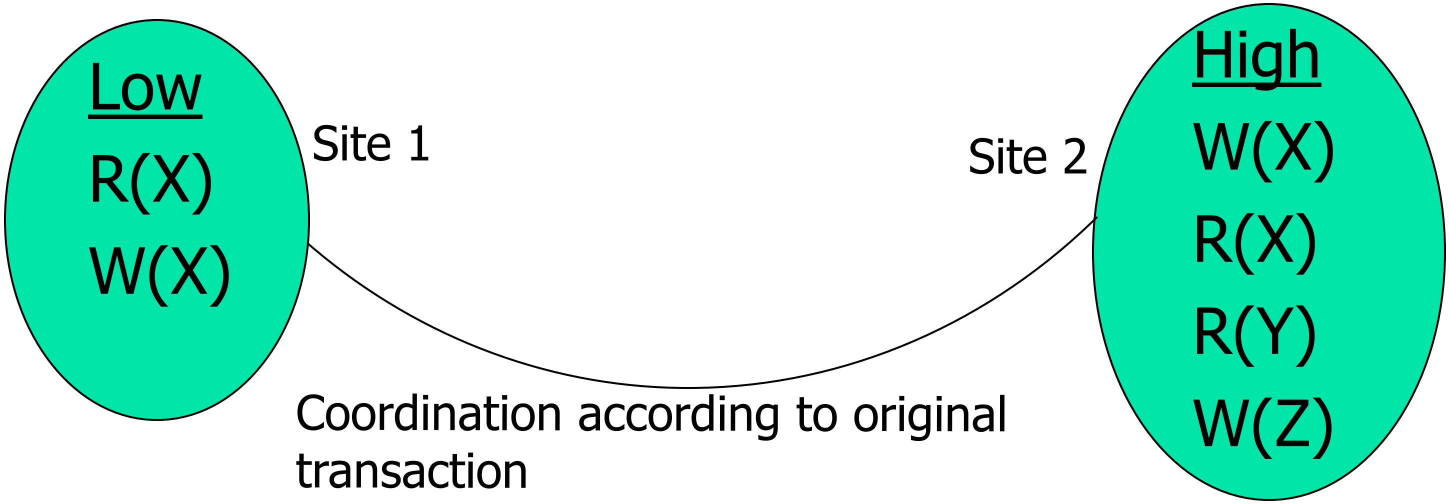
High: W(Z)

Low Labeled X data item is labeled high

High Process must read X in order to write high data item Z

Methodology

- MUSET divides the operations into single level sets or sections



Coordination according to original transaction

Correctness Criteria

- Goal

  Execute each section while achieving atomic,consistent,isolated and secure execution of the original transaction on single site.

- Criteria
- A-correctness → Fully atomic. Each commit or none
- C-correctness → Conflict equivalent to original transaction.
- I-correctness → Conflict equivalent to a serial ordering.
- S-correctness → No interference.

- Drawback:
- Except security all are achievable in multilevel transactions.

Protocols

- Multilevel Transaction Schedule(S_{Ti})

Total order of operations with ordering relation $<S_{Ti}$ such that;

- Events should be precommit,abort,commit operations
- At each level there should be abort or commit ,but not both
- Commit and abort comes after all other events
- Precommit, if exists should come before commit or abort
- Execution must preserve the order of conflicting operations with in a level.
- If read at high level follows a lower level write of same data,read must follow the commit of write.

Multilevel Schedule

- S_T over a set of multilevel transactions T1,T2,--,Tn is the ordering of all operations in S_{T1}, S_{T2}, S_{T3},... S_{Tn}, with ordering relation $<S_T$,such that it preserve the ordering of all operations in each S_{Ti}

Multilevel Execution Protocol (P)

Transformation of a set of transactions T into a schedule S_T.

$$P(T)=S_T$$

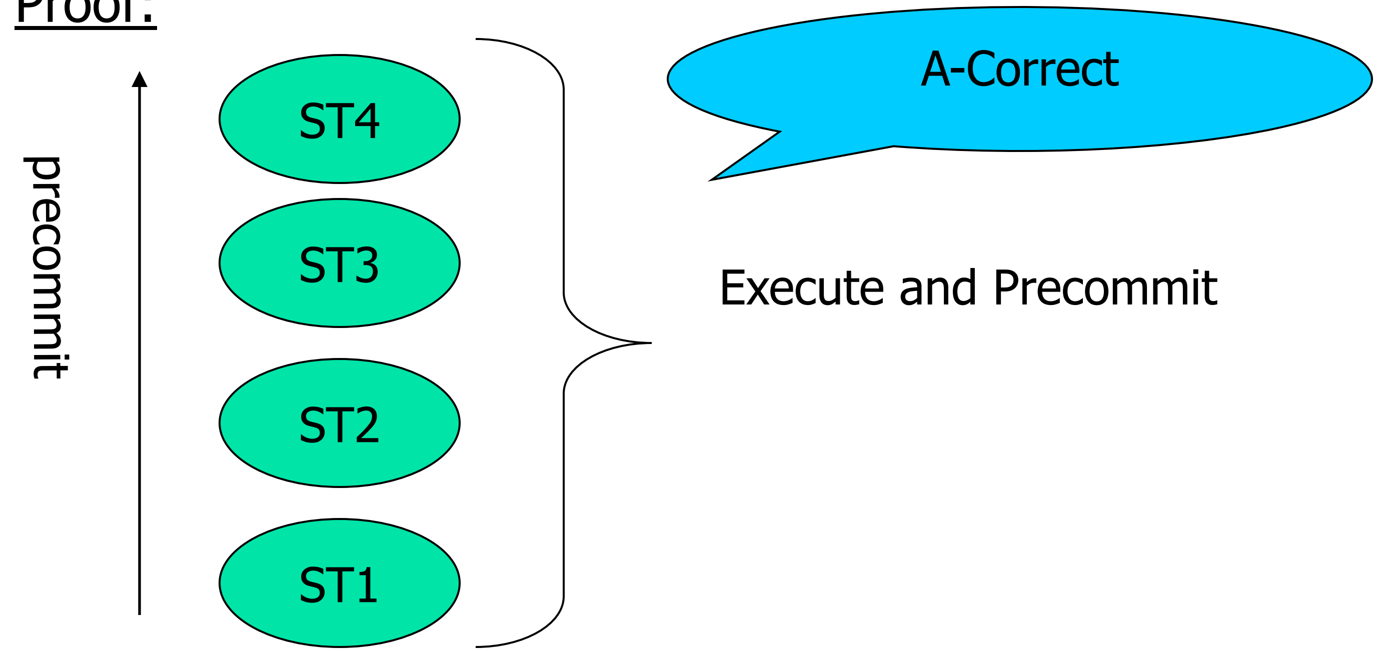
A protocol P is A-correct,I-correct,C-correct and S-correct iff for every set T of multilevel transactions, P(T) is A-correct,I-correct,C-correct and S-correct respectively.

Interactions of Correctness Criteria

- Atomicity and Security

No protocol can be both A- and S-correct.

Proof:



Atomicity and Security (cont’d)

- Precommitting in ascending order is a timing channel,because commit of lower operations is being delayed due to the execution of higher level operations.

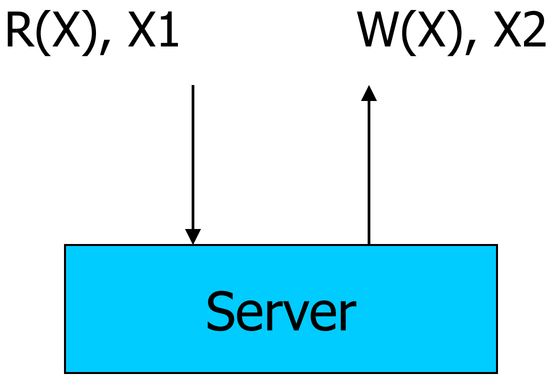
“So it is guaranteeing the A-correctness ,but not s-correctness”

Consistency & Security

- Scenario:

  High level read R(X) reads the value x1 for x and then low level write W(X) writes value x2.

This is violating S-correctness.So reorder.But now wrong version of value would be read by high level.This is conflicting C-correctness.



R(X), X1

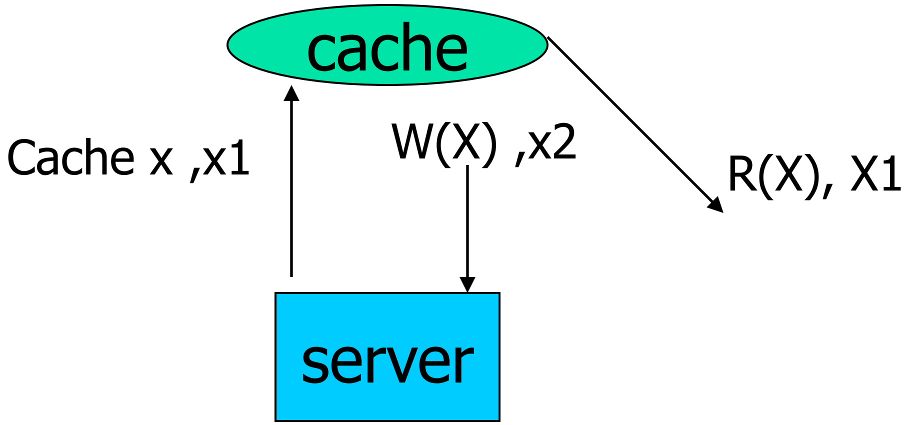
W(X), X2

Server

Consistency & Security (cont'd)

- Solution: USE OF CACHING

The value x1 is cached before updating to x2. So instead of reading x2,high level read would read x1 from cache.



Isolation & Security

- Isolation by 2PL:

  <u>2PL Locks:</u>

  Intralevel locks: All read and write locks within a level.

  Interlevel locks: Read locks covering a high read of a low datum.

  Intralevel locks→on each security level

  Interlevel locks→on “read down” operation.

Security hazard!
In High reads, low writes locks are prohibited

Isolation & Security (cont’d)

- Solution: SOFT LOCKS

  High read lock is broken when low level transaction requires lock.

•Unlock can occur unpredictably either before or after execution of read ,which is a threat to I-correctness.

•So, no protocol that uses 2PL can be I- and S-Correct

Execution Protocols

- Low Ready wait 2PL (ACIS` correct)
- Low First Multiversion Time stamping Ordering(A`CIS correct)
- Low First Hybrid Multiversing (A`CI`S correct)---used by Oracle

Transaction Processing using Agents

- What is a Agent:

Autonomous entity that can meet required goal without the assistance of a human.

- Focus of Paper:

Interaction of agents in a multi agent system by applying transaction abstraction to ensure integrity

Properties

- ACID properties are not satisfied in dealing with agents.

- Isolation is violated in two agents acting on same resource.

- Atomicity is violated when attempting to get a quote for an airline reservation among different companies,because they will not let you run commit protocols.

New Properties

- Spheres of Control

  Socs attempt to contain the effects of an action as long as there might be a necessity to undo them

  - Process atomicity → All or none
  - Process control→ control given to process so that other process cannot modify it
  - Process commitment→identifies a function which determines the modified value of each data item.

Spheres of Commitment

- Once an agent makes an commitment it is obligated to keep that commitment
- Under presence of a witness,governing all parties at events
- The committee (bearer) could discharge from it
- The committer accepts request of discharge
- The committer could discharge the committee

Society

- Witness→ Top level
- Committer→ middle level
- Committee→bottom level

Aglets

- They are used to implement the agents because they allow
- Freedom to create mobile and collaborative agents which has the ability to transport themselves throughout the network to meet specific goal.

Future Work

- Study of real cases.
- Which properties can be effectively ignored or relaxed
- Implementation of protocols
- Development of protocols capable of executing transactions over a set of DBMSs using significantly different approaches to features such as concurrency control.

Conclusion

- The inherent limitations of mobile computing systems present a challenge to the traditional problems of DBMS
- Security requirements of multilevel transactions conflict with other properties.
- ACID traditional properties can be transformed to SOCS with commitment rules to effectively perform transactions ,by using agents.