... towards Pervasive Computing

- **Gartner Group:** "by 2003 more than 137 million business users will be involved in some form of remote work"

- **Accenture:** "by 2005 over 500 million mobile devices will offer Internet access"

**Conclusions**
- Mobile hosts (MH; laptops, palmtops, smart phones, etc.) outstrip fixed hosts (FH; personal computers, desktops, etc.)
- The way information is created and processed will change within this increasingly ubiquitous network

**Need**
- Infrastructure to coordinate concurrent information access and processing in the presence of mobile hosts and users
### Characteristics of Mobility

- **Mobile information sources and consumers**
  - Physical access point to the network may change: sources as well as consumers may move.
  - Sources as well as consumers may be disconnected.

- **User and Context Awareness**
  - Tracking/monitoring information sources and consumers.
  - Consumer's information needs may shift with location change.

- **Data management techniques have to be revisited**

<table>
<thead>
<tr>
<th>Resource Limitations (Bandwidth, Memory, Computing Power, ...)</th>
<th>Optimization + Careful Resource Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td></td>
</tr>
<tr>
<td>Correctness Concerns</td>
<td>Transactional Guarantees</td>
</tr>
<tr>
<td>Combining Many Sources</td>
<td>Data Integration</td>
</tr>
</tbody>
</table>
Movements and Disconnections

- **MH movement**
  - Mobile Support Station
  - Laptop
  - Database

- **MH disconnection**
  - Mobile Phone (WAP)
  - Disconnected Palmtop
  - Database

- **MH reconnection**
  - Mobile Support Station
  - Laptop
  - Database

- **User/Workspace movement**
  - Fixed Network
  - Fixed Host
  - Database Cluster

- **FH disconnection**
  - Disconnected Fixed Host
  - Database

- **FH reconnection**
  - Database

Movements and disconnections are indicated by arrows and labels, such as "MH movement," "MH disconnection," "MH reconnection," and "FH disconnection." The diagram shows the connectivity and movement between different devices and systems.
Abstractions

Abstraction of Data Storage
(Relational) DBMS

Abstraction of Concurrency & System Failures
DBMS & TP-Monitors with Concurrency Control & Recovery

Abstraction of Method Implementations
Object-relational DBMS with Object Methods, Triggers & Stored Procedures

Abstraction of Distribution, Heterogeneity & Autonomy
Distributed & Federated DBMS, Data Integration, Conflict Resolution

Abstraction of Movements & Disconnections
Mobile DBMS, Context Maintenance, Replication & Synchronization, Profiling
Transparency of Mobility

- **Challenge**: information access and processing everywhere and at anytime while supporting
  - transparent disconnections and
  - transparent movements of users and information components

- How much transparency is indeed needed and reasonable?
• Logically related set of operations executed under certain guarantees, e.g.,
  - atomicity
  - consistency
  - isolation
  - durability

• Advanced transaction models rely on multi-tier transactions
  - increase parallelism by exploiting application semantics
  - parent-child relationships

• Mobile Transactions involve execution/initiation on MH
  - ACID cannot be supported generally
  - nevertheless: certain transactional guarantees shall be ensured always and everywhere
Mobile Transactions

Global Subtransactions

Local Transactions
Disconnection-resistant Transactions

Operation 1
Operation 2
Operation 3

Start Transaction
& Execute
Operation 1

Execute
Operation 2

Disconnect
& Execute
Operation 3

Commit
Reconnect
& Sync
Operation 1

Data base
Database
Fixed Host
Fixed Host
Mobile Support Station
CELL

Disconnection-resistant Transactions

Data base
Laptop
Start Transaction
& Execute
Operation 1

Execute
Operation 2

Disconnect
& Execute
Operation 3

Commit
Reconnect
& Sync
Operation 1
Disconnections

- "The flight worker": Working in the "intended" disconnected mode requires some lazy replication techniques
  - updates are precommitted locally
  - precommitted updates are propagated asynchronously when reconnected to the network
    - conflicts may occur
    - conflict resolution when a conflict arises

- Conflict detection via timestamps, version vectors, etc.

- Conflict handling
  - optimistic (resolution): function-based, manual
  - pessimistic (avoidance): primary copy, ROWA, quorum
Movement-resistant Transactions

MH movement

join

leave

Execute Operation 3

Commit Operation 3

hand-off

Operation 3

Operation 1

Operation 2

Operation 1

Operation 2

Commit Operation 3

Start Transaction

Start Transaction

Execution

Execution

Start Transaction
• "The train/tram/bus worker": Working while physically moving requires transparent support of cell migration and "unintended" disconnections
  - create subtransactions on several mobile support stations
  - coordinate these subtransactions correctly
  - wireless communication and cost issues

• "The home worker": Resume and/or continue work at another host (mobile or fixed)
  - continue transactions
  - create new subtransactions (within the existing workspace/transaction sphere)
Issues of Mobile Transaction Models

- MH may have transaction processing capabilities
  - MH may run transactions locally
  - MH may only initiate transactions on FH

- MH may change location and network connection while transactions are being executed
  - split computations

- MH may disconnect while transactions are being executed
  - long-lived processes
  - data replication
Characteristics of Advanced Transaction Models

- **closed vs. open**
  - Child’s results are visible to the parent only or to all?

- **vital vs. non-vital**
  - Parent’s commitment depends on the child’s commitment?

- **dependent vs. independent**
  - Child’s commitment depends on the parent’s commitment?

- **substitutable vs. non-substitutable**
  - Does there exist an alternative transaction?

- **compensatable vs. non-compensatable**
  - Are the results semantically undoable?
An Example of a Mobile Transaction Model: Kangaroo Transactions

- On disconnect: JTs and Ts can complete but no more Ts are created
- On reconnect: the T resumes
- Hand-offs: new JT runs on the new MSS (transaction split)
An Example of a Mobile Transaction Model: Kangaroo Transactions

- Builds on open nested and split transaction models
- Supports mobility and disconnections
- MH starts a Kangaroo Transaction (KT)
  - A Joey Transaction (JT) is started at the connected MSS
  - T run on the FH (as open nested transaction)
  - If the MH changes location, the previous JT is split and a new JT runs on the new location MSS
  - JT1 can commit independently from JT2
- If JT1 fails:
  - Compensating mode: undoes the entire KT
  - Split mode:
    - Previously committed JTs are not compensated
    - No new JTs are initiated
    - Current child transactions are committed or aborted upon decision of the local DBMS
# Overview of Mobile Transaction Models

<table>
<thead>
<tr>
<th>Subtransaction Types &amp; Mobility Support</th>
<th>Open</th>
<th>Closed</th>
<th>Non-vital</th>
<th>Independent</th>
<th>Substitutable</th>
<th>Compensatable</th>
<th>Temporal</th>
<th>MH Disconnection</th>
<th>MH Movement</th>
<th>MH Usage</th>
<th>User Profiling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting-/ Co-Txs Chrysanthis 93</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Isolation-Only Txvs Lu &amp; Satyanarayan 94</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDSTM Txvs Yeo &amp; Zaslavsky 94</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weak/Strict Txvs Pitoura &amp; Bhargava 94</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kangaroo Txvs Dunham et al. 97</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pro-Motion Walborn &amp; Chrysanthis 97</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>H</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Toggle Txvs Dirckze &amp; Gruenwald 98</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moflex Txvs Ku &amp; Kim 00</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>L</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
An Example of a Commercial Mobile Database Approach: IBM DB2 Everyplace

- **DB2 Everyplace**, relational database residing on the MH
- **DB2 Everyplace Sync Server**
- **Master database on the FH**

- Replication is asynchronous and on demand
  - entire tables, subsets of columns/rows, views, joins, unions

- **Synchronization by publish and subscribe**
  - Refresh propagation: all data are sent to subscribers
  - Incremental propagation: only changes are propagated to the subscribing MHs

- Conflicts are handled by checking the version of each record in each table in the replication subscription

- No savepoints, No transaction nesting
### Overview of Commercial Mobile DB Approaches

<table>
<thead>
<tr>
<th>Subtransaction Types &amp; Mobility Support</th>
<th>Savepoints</th>
<th>Tx Nesting</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM DB2 Everyplace</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Informix Cloudspace</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft SQL Server CE</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Oracle Lite</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Sybase Anywhere</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- only a few supports nested Txs (closed, vital, dependent)
- no compensating / alternative Tx
- "basic" data replication and synchronization techniques
- no transaction mobility (hand-offs)
MH stores data as records in PalmDBs

- each PalmDB is associated with an application
- each record has a set of status bits indicating whether the record has been created, modified, or deleted since last sync

FH maintains its own copies of the PalmDBs, including its own versions of the status bits

- also maintains a snapshot of each PalmDB taken immediately after most recent sync
- runs HotSync Manager

MH initiates synchronization (fast vs. slow sync)

Problem: No support for synchronizing with multi-user concurrent data sources

- no notion of “interest” in a subset of the records in a database
- no notion of transaction at all
Industry Consortium with most major players: Ericsson, Nokia, Motorola, Palm, Psion, IBM, …

Goal: enabling cross-format, cross-system synchronization

Simple architecture
  - MH is intermittently connected und FH is continuously available

Consists of a standard set of message types, each represented as an XML document

Conflict resolution is dealt with abstractly
  - only standard status codes to implement typical policies are given

<table>
<thead>
<tr>
<th>Two-way</th>
<th>MH and FH exchange only modified records, MH sends first</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow-sync</td>
<td>MH sends all records</td>
</tr>
<tr>
<td>One-way, MH only</td>
<td>MH sends only modified records to FH</td>
</tr>
<tr>
<td>Refresh, MH only</td>
<td>MH sends entire DB to FH</td>
</tr>
<tr>
<td>One-way, FH only</td>
<td>FH sends only modified records to MH</td>
</tr>
<tr>
<td>Refresh, FH only</td>
<td>FH sends entire DB to MH</td>
</tr>
<tr>
<td>FH Alerted</td>
<td>FH initiates sync</td>
</tr>
</tbody>
</table>
• Mobile transactional coordination has to deal with
  - weak connectivity and frequent disconnections
    • asynchronous, dynamic replication with profiling
    • publish & subscribe for data recharging & propagation
  - large-scale replication
  - user interaction / feedback
  - long-running tasks and decentralized commitments
  - real-time constraints

• Commercial mobile database approaches mostly neglect the latter issues
Open Questions

- Where to implement the abstraction of disconnections and movements?
  - Do we really need extensions to transaction models?
  - Can we model mobility issues as additional steps of an overall transactional process?
  - Which part is only "network staff"?
  - ...

- Design of mobile transactional applications
  - How to appropriately express mobility issues?
    - connection duration and costs
    - disconnection times
    - data and function interests
    - consistency requirements
  - ...