Summarisation for Mobile Databases[1]

Darin Chan and John F. Roddick

School of Informatics and Engineering
Flinders University of South Australia
P.O.Box 2100, Adelaide 5001, South Australia
{darin.chan,roddick}@infoeng.flinders.edu.au

In mobile computing, issues such as limited resources, network capacities and organisational constraints may cause the complete replication of large databases on a mobile device to be infeasible. At the same time, some on-board storage of data is attractive as communication to the main database can be inconsistent. Thus, as the emphasis on application mobility increases, data summarisation offers a useful solution to improving response times and the availability of data. These summarisation techniques can also be of benefit to distributed databases, particularly those with mobile components or where the profile of the transaction load varies significantly over time. This paper surveys summarisation techniques used for mobile distributed databases. It also surveys the manner in which database functionality is maintained in mobile database systems, including query processing, data replication, concurrency control, transaction support and system recovery.

Classification: H.2 (Database Management), H.3.2 (Information Storage).

1. INTRODUCTION

In contrast to distributed databases operating on traditional hardware, the devices available for mobile databases may exhibit severe resource limitations which can affect the manner in which data is managed. However, in many respects both conventional distributed databases and mobile databases may be considered as special cases of a common architecture and both can borrow ideas from the other (Dunham and Helal, 1995). Indeed, Holliday, Neumann and others (Holliday, Agrawal and Abbadi, 2000; Neumann and Maskarinec, 1997) discuss the addition of mobile components within a distributed database system. In short, many distributed data management issues are similar to those that affect the mobile database systems, and many innovations being considered for mobile databases are equally applicable to some distributed data management problems. This paper surveys some of these techniques focussing on summarisation techniques applicable to mobile databases. To do this we also discuss the manner in which database functionality is maintained in mobile database systems, including critical functions such as query processing, data replication, concurrency control, transaction support and system recovery.

The paper is structured as follows. The remainder of this section will provide a motivation for this paper. Section 2 will then discuss issues concerning databases used in a mobile environment

[1] We gratefully acknowledge support from the Australian Research Council through SPIRT Grant C00107533, and our partner HP Labs.

Manuscript received: 30 September 2004
Communicating Editor: Sidney A. Morris

and, for completeness, Section 3 discusses the three major classes of wireless technology. Section 4 will then examine the architectures available in a mobile environment in terms of the operational modes experienced by mobile applications, the wireless technologies currently available and the database architectures that may be used. Section 5 then discusses data summarisation techniques and Section 6 explores different techniques for implementing common database functions. The paper concludes with a discussion of outstanding issues.

1.1 Motivation

Effective data management has been identified as a significant issue in mobile application development and uptake (Dong and Mohania, 1996; Pitoura and Samaras, 1998). A good example of an application of mobile databases is the medical profession, where within a hospital or medical practice scenario, practitioners often find themselves moving between patients and locations. Using mobile systems, it is possible to provide, in an effective manner, updated data to the medical practitioner without the need for physical connection. The mobile system then enables the practitioner to access current data on the patient for diagnostic purposes, and to update details about the condition of, and recommended actions for, the patients that they are monitoring. Additionally, outside the hospital or the medical practice, practitioners visiting patients may also benefit from the use of mobile databases. Before leaving, the practitioner's database may be summarised to maximally contain the data that is required, based on a variety of parameters that will change dynamically. Once the practitioner leaves, connection to the main database will be expected to be weak or nonexistent. The practitioner will then need to rely on the data available locally. Any added patient information would then be stored on the mobile computer and merged back with the main database when the connection is reliable. To do all this, a sophisticated and robust mobile data management system must be in place, the characteristics of which are the subject of this paper.

2. MOBILE DATABASE ISSUES

There are many issues concerning the effective use of mobile systems, both in respect of the current technology and those likely to become available in the near future. For mobile databases the most important of these are:

- the relative unreliability of connections (and the variability of bandwidth when connected),
- the limitations on storage capacity, and
- the security and privacy issues created when a computer is in a mobile environment.

This paper will focus only on the first two of these issues. Issues concerning security and privacy may cause significant problems in regards to databases in mobile environments and may include the identification and authentication of users and the theft of data during transmission (Heuer and Lubinski, 1996; Zaslavsky and Tari, 1998). However, these issues are outside the scope of this paper.

2.1 Connectivity and Disconnections

An important limiting characteristic of mobile computers is their finite battery capacity and the low communication bandwidth available and/or affordable (Imielinski and Badrinath, 1994; Lubinski, 2000; Pitoura and Bhargava, 1993). These limits may lead to frequent disconnection due to both the need to reduce connection costs and because of technical limitations and failure (Heuer and Lubinski, 1996; Lubinski, 2000).

Advancements in battery technology have allowed longer intervals between recharge[2]. However, even with the longer life, there are higher energy costs involved while a mobile computer transmits or receives data (Pitoura and Bhargava, 1993; Zaslavsky and Tari, 1998), and this may rapidly decrease the life of the battery. Thus, intentional disconnection may be required through power management software to allow higher priority operations to continue. In addition, in most situations there are likely to be many mobile users connected to their centralised or distributed database servers and in some locations, wireless coverage may be patchy. The available communication bandwidth in a mobile environment must therefore be shared between each user, thereby restricting the bandwidth available. As a result, networks may experience outages that may cause the mobile device to be subjected to frequent disconnection. Thus, in terms of mobile databases, access between central servers and the mobile system may be unreliable at times (Madria, Mohania and Roddick, 1998). The disconnections that may occur can disrupt a query being performed and as a result, prevent or slow query response.

2.2 Storage Capacity

Another of the more important resource limitations of mobile devices is their limited information storage capacity (Lubinski, 2000; Pitoura and Bhargava, 1993). Large and multiple hard drive systems, such as those found in common desktop and server computers, cannot be accommodated into a mobile computer, given that they are required to be portable.

Improvements in hard drive technology have made modern hard drives smaller and with a greater capacity. Devices holding 60Gb are now commonly available for laptops and, while comparable to single drive desktop systems, commonly have a much smaller capacity compared to the centralised servers that support corporate databases. However, even with the reduced physical size of the modern hard drive, most are still as large as a PDA. Consequently, the storage capacities of even laptop-based hard drives are about 500 times more than that utilised by a flash disk emulator[3]. These flash disk emulators may provide attractive energy consumption and performance, but are still relatively expensive and have limited capacity (Douglis, Kaashoek, Li, Cáceres, Marsh and Tauber, 1994). Due to the storage limitation of mobile computers, it is thus difficult (even if it was logically desirable) to create replicas of large databases on such devices, particularly on handheld devices such as PDAs, which are the current focus of our work.

3. MOBILE WIRELESS TECHNOLOGIES

In contrast to most distributed networks, the nature of the communications link determines to a much greater extent the appropriate protocols to be adopted for data management. We therefore briefly describe the three major classes of wireless technology based on their operating range.

The most local is the wireless personal area network (WPAN). This describes wireless networks that have a small transmitting range, commonly less than 10 metres. As a result of this short range, they usually have low power consumptions. An example of a WPAN technology is Bluetooth – an open standard for shortwave radio communications that allows two-way connectivity of up to 1 Mbps. The concept of WPAN is based upon proximity networking. That is, devices may come and go frequently and devices may join or collaborate with each other or other networks in proximity as the WPAN user moves. The current Bluetooth technology allows any devices with Bluetooth to automatically synchronise with each other if they are within proximity of each other.

[2] In particular, PDAs (such as the iPaQs produced by HP) provide 600-1400 mAH Li-Ion batteries that may last up to 10–14 hours of continuous usage, while the Lithium-Ion battery packs for laptops may provide continuous usage for 2–10 hours depending on the battery configuration of the device.

[3] Currently, a typical capacity for laptop is around 60Gb while flash disks may provide around 128Mb. As technology has improved, this ratio has been maintained.

At the next level, wireless local area networks (WLAN) are a wireless extension of the traditional LAN (Imielinski and Badrinath, 1994). Like the LAN, they may provide medium coverage from ten to a few hundred metres and relatively high data rates. Common examples for WLAN are the IEEE 802.11b and 802.11g standards that provide data rates up to 11 Mbps and 54 Mbps, respectively. As a result of the medium range and high data rates, the consumption of energy is relatively high and only portable systems such as laptops hold the required energy to keep a reasonable battery life and be able to continuously support such networks. WLANs work more efficiently when used in networks that are stationary. That is, mobile units are restricted to the coverage of the wireless access point, and roaming between networks is usually difficult.

Finally, the wireless wide area networks (WWAN) covers network systems capable of providing services, which includes voice, data and video, to geographically large areas – commonly kilometres. The first generation of wireless networks were designed mainly for voice communications and utilised analog technology such as FM modulation. The second generation of wireless networks utilise digital transmission to provide voice and limited data services. This is the current and commercially accepted generation of wireless network, and it includes systems such as the Global System for Mobile communication (GSM), CDMA networks and paging networks. The third generation of wireless networks is evolving from mature second generation systems and is currently beginning to appear on the commercial market. Its aim is to provide a single set of standards for high voice, data and video communication for a wide range of wireless applications.

These technologies are generally terrestrial and employ base towers to provide wireless communications to mobile computers. However it is also possible to use satellites to relay the communications. The Iridium System is one such system that employs 66 low-earth-orbit satellites that are networked together to provide wireless voice and paging coverage to mobile devices. Leopold *et al* (1993) provide more details about the Iridium System.

4. MOBILE DATABASE ARCHITECTURE

A typical architecture for a mobile database includes a small database fragment residing on the mobile device derived from the main database, see Figure 1 (Madria *et al*, 1998). This architecture may be considered a client/server architecture in which the main database is located with the server and the summary database resides within the client or mobile device, and has been used in a variety of projects (Demers, Petersen, Spreitzer, Terry, Theimer and Welch, 1994; Phatak and Badrinath, 1999; Chan and Roddick, 2003; Madria *et al*, 1998). Indeed, the Bayou architecture, as discussed in Demers *et al* (1994), uses a client/server architecture with the addition of lightweight servers to allow certain clients to serve data in their caches to other clients. Such a technique allows for better availability of data but has a weak consistency property that may require complex conflict detection and resolution.

Other database architectures have also included an agent within the structure, ie. a server/agent/client architecture. The role of the agent may vary between different implementations and may reside either in the server or in a mobile device. A bandwidth dependent agent, termed an intermediary, is described by Zenel and Duchamp (1995) to allow only essential data to travel through a slow communication link, while the other data are filtered out and delayed. Heuer and Lubinski (1996) introduce an information agent to intelligently handle the transfer of multimedia specific data, while Lauzac and Chrysanthis (1998a; 1998b) also introduce an agent, called a view holder, to provide individual updates to the client's database.

In the examples above, the client/server network architecture discussed describes a fixed topology network in which the server is fixed and mobile computers may move from one server to

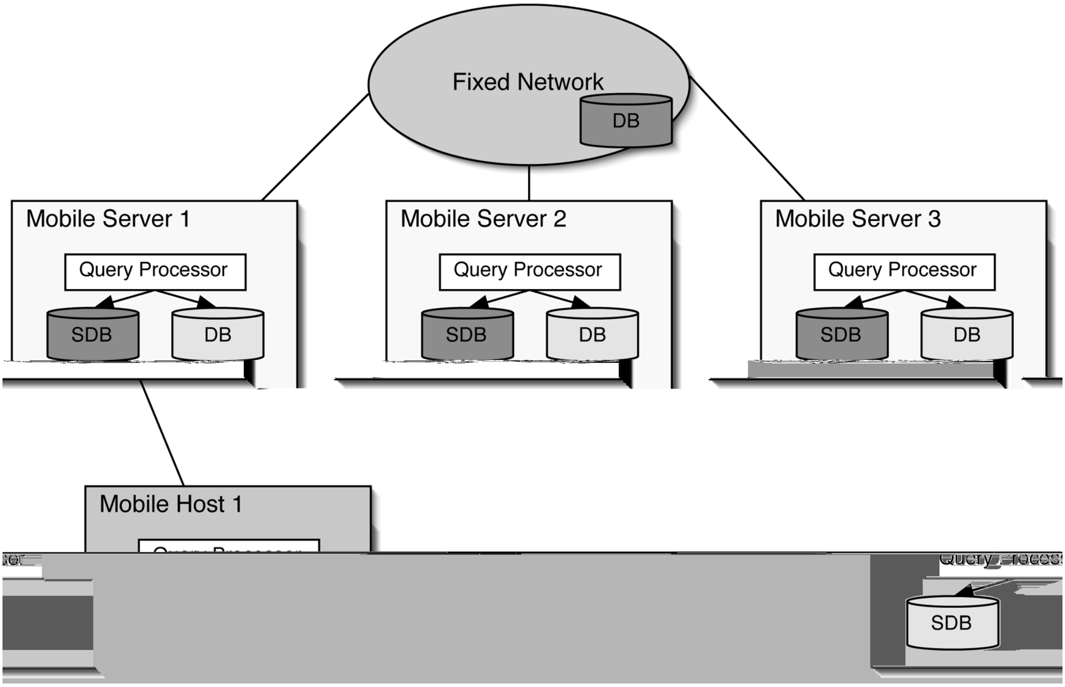


Figure 1: Movement of Mobile Database between Servers

another if they wish. The ad-hoc network system discussed by Fife and Gruenwald (2003) proposes a wireless system that may change frequently in topology as the nodes in the area organise themselves. These nodes consist of both clients and servers, which are both wireless and mobile in comparison to the fixed servers mentioned earlier. Such changes in network topology results in the possibility of nodes being in a standalone network or part of a larger network. An example of *ad-hoc* networks is the Bluetooth scatternet. A scatternet is a collection of piconets that are connected together, where a piconet comprises of a master node that coordinates with all the other proximate mobile nodes (Law and Siu, 2001). An *ad-hoc* network may lead to a peer-to-peer architecture, in which clients may also communicate with other clients to share the data that they are holding. This will then result in a higher availability of data but may compromise consistency if clients are allowed to update any data.

4.1 Mobile Operation modes

Due to the restrictions on their resources, there are several modes of operation that the mobile systems may experience, as compared to non-mobile systems (which are either fully connected or disconnected). Indeed many non-mobile systems are written assuming only a fully connected status and are rendered unavailable otherwise. Mobile systems do not normally have this luxury and special protocols are required to handle each of these modes (Neumann and Maskarinec, 1997; Pitoura and Bhargava, 1993). The three modes of operation that are of interest here are:

- **Full connection mode**
  In a fully connected mode, the mobile computer is continually connected to a server. In this mode hand-over protocols would be required if a cellular structure is used and when mobiles

move from one cell to a different cell. The hand-over protocol may involve a new communication link between the mobile unit and the new server, and saving and transferring the states from the old to the new server (Madria *et al*, 1998). The communication hand-over to a new cell should be transparent to both users and applications not specifically involved with the hand-over process.

- **Disconnected mode**
  In this mode, the frequent disconnections of a mobile host, as described in Section 2 must be addressed. One possible solution to deal with such disconnections would be the use of a proxy for the mobile computer (Stanoi, Agrawal, El Abbadi, Phatak and Badrinath, 1999). This would ensure the query continues to run even when the mobile component is disconnected, and the mobile unit may request an update from its proxy when it does reconnect. In addition, mobile computers may voluntarily move into this disconnected mode when idle or low on battery power, to free up bandwidth resources and extend battery life (Pitoura and Bhargava, 1993). While operating in disconnected mode, any applications that had used the communication link before the disconnection would be required to save their current communication state, and where possible continue with its other processes. Upon reconnection and depending on the saved communication states, applications may resume transmission or reception, or retransmit a request to begin the communication anew.

- **Partial or weak connection mode**
  In this mode, the mobile unit is connected to the rest of the network through low or intermittent bandwidth. The degree to which the communication bandwidth is available may vary from marginally less than full bandwidth availability to almost no connectivity. Partial or weak connection mode may occur when the mobile units are in shadow areas within or on the edge of a cell, where reception is poor. The partial-connection protocol would then be required to allow the mobile client to limit its communications to the network. Applications that use the communication link may then experience longer transmission time and may be required to extend timeouts to anticipate a lengthy response time.

The protocols to handle the above-mentioned modes may also be deliberately invoked by the mobile unit to allow conservation of its limited resources. Research has generally focussed on the disconnected and partially connected operation modes (Keunning and Popek, 1997), since in these modes the mobile device must rely more heavily on its own resources to process any transactions.

5. DATA SUMMARISATION

To date, many database summarisation techniques used on traditional databases involve the use of structural reduction techniques that reduce the volume of data without considering the use (i.e. the importance to the user) of the data. However, in mobile distributed environments context sensitive information is important in that it may provide more relevant information to users readily and efficiently (Hawick and James, 2003; Jones and Brown, 2004; Covington, Long, Srinivasan, Dev, Ahamad and Abowd, 2001; Chan and Roddick, 2003). Context sensitive data is information that has a semantic connection to user of those data. This information may then be used in the size reduction of the database, hopefully maximising the relevance of the data stored and thus the usability of the summarised database (Chan and Roddick, 2003). This section explores both context sensitive and insensitive techniques used for traditional, distributed and mobile databases.

5.1 Schema Modification Techniques

These techniques modify the schema by removing either attributes or tuples from the relations of a database to produce new relations that are reduced in size. There are three techniques that may be included as schema modifying techniques.

- **Projection (vertical) methods**
  Simply used, this method involves the vertical elimination of attributes in a relation within the database (Ceri, Negri and Pelagatti, 1982; Lubinski, 2000; Roddick, Mohania and Madria, 1999), and is one of the more commonly used techniques for reducing the size of a database. The method of projection may also be utilised in conjunction with other types of summarisation techniques, such as concept hierarchies. Projection provides a modification to the structure of the database schema. That is, the technique is used to remove attributes within the base relations.
- **Selection (horizontal) methods**
  The selection method involves the horizontal reduction of tuples within a database (Chu, 1992; Cornell and Yu, 1990; Muthuraj, Chakravarthy, Varadarajan and Navathe, 1993; Navathe, Ceri, Wiederhold and Dou, 1984; Navathe and Ra, 1989; Lubinski, 2000; Roddick *et al*, 1999) commonly through a selection of values of a specified attribute. Again, selection may also be found in conjunction with other summarisation techniques.
- **Hybrid**
  The two techniques above may be combined to produce materialised views. There are different schemes by which hybrid fragmentation may be achieved. For example, if we first perform horizontal fragmentation followed by vertical fragmentation then subsequent rejoining of the fragments (assuming each fragment is disjoint) vertically, will produce the horizontal fragment

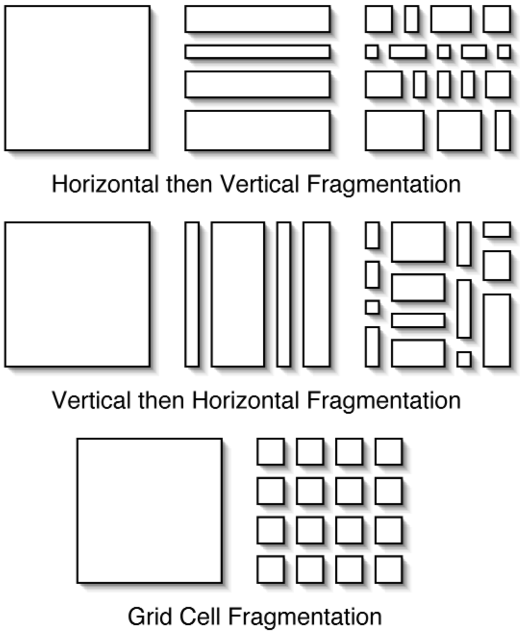

Horizontal then Vertical Fragmentation
Vertical then Horizontal Fragmentation
Grid Cell Fragmentation

Figure 2: Hybrid Fragmentation Methods

(See Figure 2). Horizontal joins, however, will not be possible since the fragments may not contain the same attributes. The same occurs with hybrid fragmentation where vertical fragmentation is performed before horizontal fragmentation, with the exception that vertical joins are not possible and horizontal joins will produce the original vertical fragment (See Figure 2). The hybrid fragmentation method described by Navathe, Karlapalem and Ra (1996) is independent of the sequence that the fragmentation is performed. The fragments, termed grid cells, have the property of being able to be both vertically or horizontally joined. This is possible since each grid cell belongs to exactly one horizontal and one vertical fragment.

Views define a function from a subset of relations to a derived relation, and may be materialised by physically storing the tuples of the view (Lauzac and Chrysanthis, 1998b). The use of views in terms of mobile systems has largely been discussed within data warehouse applications to allow maintenance and updates of the mobile client's database (Lauzac and Chrysanthis, 1998a; 1998b; Stanoi *et al*, 1999). Lauzac and Chrysanthis' materialised view holder discussed earlier acts as a proxy during network disconnection to provide the required updates and views to the mobile client upon its reconnection. Note that since views are predetermined queries, an equivalent query employing rewriting techniques may also be used to allow query optimisation (Halevy, 2001).

Schema modification techniques by themselves are usually insensitive to the use of the data. However, they may also be used with criteria that are sensitive to the usage of the data such as a previous usage history, to provide techniques that are context sensitive.

5.2 Domain Modification Techniques

Domain modification changes the domain of a relation in such a way that it is possible for data to be grouped together and thus produce a reduction in size of the final result.

- **Concept Hierarchies**

  The usage of concept hierarchies to reduce the size of a database may be considered an example of a domain modification technique. This technique was proposed by Madria *et al* (1998) as a query processing model for mobile computing to facilitate data volume reduction for a summary database. The concept hierarchy process is capable of organising data and concepts in a hierarchical form by providing a mapping or generalisation of the lower layer concepts to their corresponding higher level concepts (Han and Fu, 1994). The central idea is that of concept ascension in which low level data items are elevated onto a higher level within the hierarchy and duplicates tuples are then removed. This hierarchy may be externally supplied or dynamically generated or refined, as demonstrated by Han and Fu (1994). In terms of context sensitivity, current techniques for concept hierarchies may be considered insensitive to the use of the data since they do not have the capability of identifying specific data for storage. Thus, while still a useful addition to schema decomposition, these techniques may not be determining the optimum data required by the user.

- **Abstraction through aggregation**

  This is the summarisation of selected attributes through the use of metadata by creating new data items such as the sums and averages of existing data (Heuer and Lubinski, 1998; Lubinski, 2000). Work reported by Barbara *et al* (1997) examined several techniques that may provide rapid but approximate answers for data warehouse applications. Aggregation thus allows not only a reduction of the amount of data required for storage by generalising the data into a new collection of objects, but is capable of quickly answering queries through those new objects. All data reduction ultimately results in some loss in information where an approximate form must be used to replace that data. Therefore, this technique works well when the user does not require specific or detailed data. The contribution that abstraction makes is that it generalises the data into a new collection of objects.

- **Gradual Data Reduction**
  Lubinski (2000) introduces the gradual data reduction technique as a way of reducing the data of the query results before it reaches the mobile system. This technique specifies different domains of data precision and a layered reduction for those domains. The domains of data precision relate to the interestingness that the user has placed on the resultant data through domain boundaries. For example, Domain 1 might contain data that is most relevant to the user within a specified domain boundary, and corresponds to Layer 1, where no reductions are performed. Furthermore, Domain *n*, where *n* is the final domain, consists of data that is of marginal interest. These data corresponds to Layer *n* that utilises the most extreme data reduction.

- **Surrogates**
  Finally, in situations where complex data types are used in a database, a surrogate method may be used. Data that are complex may usually lead to extra resources being required to access, transfer and present them (Lubinski, 2000). Thus, using surrogates for substituting complex data types with corresponding simpler data type may relieve resources for other operations. An example of the replacement technique is to substitute a large image file with its equivalent but simpler text description. The surrogate technique is more effective when used with more complex data types that have equivalent but simpler substitutes. Such a technique would modify and replace the data type of the attributes of the base relations.

5.3 Fragmentation and Data Allocation

In much of the literature, fragmentation and data allocation are considered as separate processes. As discussed earlier, many methods of fragmentation use a representative set of transactions or queries as a base to partition the global relation into different fragments. For example, attribute and predicate usage matrices are used as starting points for vertical and horizontal fragmentation procedures, respectively. These matrices are formed by examining the frequency that an attribute or predicate is accessed by the given set of transactions. While basing the fragmentation on a given set of transactions may be important, it may not encapsulate the importance of a query that a user may ask, especially when the representative set is small or out of date. Moreover, it can be somewhat blunt in its decisions for inclusion and other information, such as location, the time of events stored in the database, inductive association between data items and so on may also be useful when deciding on the content of a summarised distributed database.

In terms of data allocation, most of the literature again uses a representative set of transactions to minimise the cost of accessing and processing queries at each site, and allocates fragments accordingly. This is perhaps the same set as used in the fragmentation stage. Shephard *et al* (1995) investigate a two-phase approach to data allocation. The first step creates the fragment clusters based upon the given set of transactions and their access frequencies, the second allocates the clusters based on a divide and conquer algorithm such that the cost of processing a set of queries is minimised. These types of allocation technique are most useful when future access patterns are the same. Ahmad *et al* (2002) suggest that evolutionary techniques, such as genetic, simulated evolution, mean field annealing and random neighbourhood search algorithms, can be useful in allocating fragments to each site. Again, these algorithms are based on a representative set of queries to minimize access costs. While these algorithms were used within a traditional distributed database system, it is also possible to use similar algorithms for mobile distributed database systems where the access costs would also consider the resource limitations of a mobile computer (Huang, Sistla and Wolfson, 1994).

Wolfson and Jajodia (1995) propose a reallocation of fragments during runtime using data collected from recent transactions. Similar work was undertaken by Brunstrom *et al* (1995) who show that dynamic reallocation of fragments may significantly improve the performance of distributed databases with changing workloads. In addition, there is also research that examines fragmentation and allocation within a single process. Note that in some instances, elements of replicated fragments may exist (and potentially be updated) at more than one site.

In our previous work (Chan and Roddick, 2003) we examined an allocation method using different context-sensitive criteria, including location, time, inductive and enumerated criteria. When combined the criteria specifies through an algorithm which data are to be included in the mobile database. Similarly, changes in the user's context would induce a change of the data within mobile database.

5.4 Data Compression

There are many methods of data compression based mainly on character encoding and repetitive string matching (Graefe and Shapiro, 1991). Techniques that allow the compression of data usually aim to reduce the redundancy that may be found in stored or communicated data (Lelewer and Hirschberg, 1987). More details of the available compression techniques may be found in a number of surveys (Bell, Witten and Cleary, 1989; Lelewer and Hirschberg, 1987; Cannane and Williams, 2001). In the context of databases, projects such as those reported by Severance (1983); Cannane, Williams and Zobel (1999); Roth and Van Horn (1993); Cormack (1985); Graefe and Shapiro (1991); Ray, Haritsa and Seshadri (1995) investigate the use of data compression techniques within a database. Additionally, many techniques surveyed in those papers include compression methods that deal only with text data types. One particular method that does consider the different data types is the RAY algorithm as described by Cannane, Williams and Zobel (1999).

6. MOBILE DATABASE FUNCTIONALITIES

6.1 Query Processing and Optimisation

An important issue when dealing with queries is the way they could be processed. In previous work (Roddick, 1997), three approaches were identified.

- Firstly, to allow a query processor to determine the appropriate database to answer the query, whether it is the main or summary. This relies on in-built intelligence that can determine whether the local database can answer any arbitrary query.
- Secondly, to adopt a course grained approach where the query is sent to both databases and the first response to be received will be used. This option can be expensive but is guaranteed to produce an answer quickly.
- Lastly, to adopt a fine grained approach in which the query is segmented and those segments are run either in parallel or in series on both databases. This has the advantage of utilising the quicker local database more often and, on occasions, may obviate the need for parts of the query to be processed. The parallel option is the quicker but may result in redundant communication. The serial option uses the central database when the local database is unable to process a given sub-query and the results of that sub-query are deemed necessary.

The process of deciding which approach is the optimum in mobile distributed environment is difficult since the mobile computers are prone to a dynamically changing environment and state.

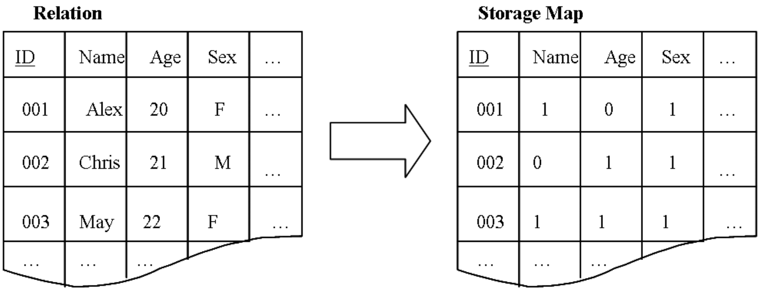
Additionally, it is important to understand the responses that may be returned when using an incomplete database such as a summary database. In Madria *et al* (1998), four types of query answers were identified as follows:

1. Complete[4] and sound[5],
2. Potentially understated (sound but may be incomplete),
3. Potentially overstated (complete but may not be sound),
4. Wrong.

The difference between 1 and 3 lies in the relaxation of the closed world assumption (Reiter, 1978) that dictates that anything not recorded within the database may be assumed false, and thus potentially overstated responses are correct if there is no evidence to prove otherwise. This can be useful as part of a fine-grained query decomposition process (Roddick, 1997).

With traditional databases, the objective is to provide query answers that are logically complete and sound. However, summary databases are by their nature associated with a loss in data and are thus incomplete but sound. Consequently, it is not always possible to provide both sound and complete answers. However, for these cases, it is sometimes possible to provide knowingly overstated responses (for example, through generalised responses). Thus, not only may query responses be the same as those given by the original database, but it may also, where allowable, provide responses that may be potentially overstated. Furthermore, in Madria *et al* (1998), the query processor is responsible for determining the correct database to answer a query depending on certain conditions such as user interaction, heuristics, the nature of the query or type of connections. Once determined, query rewriting techniques using the available concept hierarchies may be used to provide sound but not complete answers to the query. Ganguly and Alonso (1993) present three search algorithms, namely adaptation of partial order dynamic programming, linear combinations and linearset algorithm. These algorithms were used for determining energy efficient query optimisation within a mobile environment.

In our work (Chan and Roddick, 2003) we examined the use of storage maps, which are created to represent the data contained within a summary database, to answer queries. A storage map for a relation is a combination of a single binary value for each attribute values, as shown in Figure 3. This indicates which values exist in the summary database. Primary keys are usually not converted since



Relation

ID	Name	Age	Sex	…
001	Alex	20	F	…
002	Chris	21	M	…
003	May	22	F	…
…	…	…		

Storage Map

ID	Name	Age	Sex	…
001	1	0	1	…
002	0	1	1	…
003	1	1	1	…
…	…	…		

Figure 3: Storage Map

[4] In that a query on the summary database will return at least the data that the same query would return on the main database.

[5] In that a query on the summary database will return no additional data that the same query would return on the main database.

they are used as references back to the tuples in the source relation. A fine-grained approach is then adopted by using the storage map to determine which data may be extracted to answer the query.

The concept and semantics of the null value in relational databases has been discussed widely since the introduction of the relational data model in the late 1960s. With the introduction of highly mobile, distributed databases, in order to preserve the accepted soundness and completeness criteria, the semantics of the null value needs to expand to reflect a localised lack of information that may not be apparent for the global/central database. Our work also extends the notion of nulls to include the semantics of 'local' nulls.

6.2 Data Replication Schemes

Data replication schemes specify both the way that a replicated site may update the database and how an update may propagate to other sites. There are two mechanisms available that relate to where an update may be processed. The first designates a primary site through which all updates are performed and from which updates are propagated. The second uses an 'update everywhere' approach in which the update is processed at the site of origin and updates are then propagated to all sites that have an interest in the data (Wiesmann, Pedone, Schiper, Kemme and Alonso, 2000a; 2000b).

Within distributed databases, there are generally two propagation protocols through which sites may propagate an update – eager or lazy (Wiesmann *et al*, 2000b; Gray, Helland, O'Neil and Shasha, 1996). An eager update protocol involves keeping the propagation process as part of the transaction, and thus resulting in all sites being synchronised during an update (Gray *et al*, 1996). For eager update protocols, further classifications may be possible in terms of how the transaction is terminated (such as through voting or non-voting techniques) and the manner in which the servers communicate with each other (constant or linear interaction) (Wiesmann *et al*, 2000a). While eager updates work well for traditional distributed database, it is often unsuitable for mobile distributed database as frequent disconnections may cause the update to fail much of the time. Eager updates have the advantage of providing serialisability of the update execution on each of the sites and consistency between all the replicas (Breitbart, Komondoor, Rastogi, Seshadri and Silberschatz, 1999). However, it has the disadvantage of requiring higher message overheads and extending the response times as all the replicas would be involved in the update process (Wiesmann *et al*, 2000b).

Lazy update protocols, on the other hand, propagate an update only after it has been committed on the site that initiated the update (Wiesmann *et al*, 2000b; Gray *et al*, 1996; Breitbart *et al*, 1999; Ladin, Liskov, Shrira and Ghemawat, 1992). Lazy updates provide a faster response time since other replicas are only involved after a update has been committed, but global serialisability may be compromised since it is possible for two or more sites to commit the same data at the same time (Gray *et al*, 1996). Therefore, further schemes would be required to detect and resolve such conflict.

The combination of both eager and lazy scheme is also possible and was examined in Lubinski and Heuer (2000). The authors discussed a framework for a protocol to allow replication that is configured for specific mobile applications.

There is a large body of literature dealing with the different schemes, most comprising various combinations of the classifications discussed above. Many of these are also surveyed in Bernstein, Hadzilacos and Goodman (1987); Davidson, Garcia-Molina and Skeen (1985); Wiesmann *et al* (2000a); Wiesmann *et al* (2000b).

6.3 Concurrency Controls

Concurrency control mechanisms are used to ensure consistency within a database management system when multiple users access and change data. Concurrency control methods are usually dependent on where update procedures may occur. For example, in cases where updates may only occur on the central or primary site, simple two-phase locking management on that site may provide the required consistency. For a distributed database system, where updates may occur at any site, a voting method is used in which the site performing the update requests a lock from all relevant replicas (Elmasri and Navathe, 2000). This would then lock the data for the duration of the update. In mobile distributed database systems, however, such locking procedure may be costly in terms of an increase in the number of messages that are required to coordinate the locking and unlocking of data while the mobile site is moving from one server to another (Jing, Bukhres and Elmagarmid, 1995). An Optimistic Two Phase Locking for Mobile Transaction (O2PL-MT) was proposed in Jing *et al* (1995) to reduce the number of messages, based on the Optimistic Two Phase Locking (O2PL) algorithm presented by Carey and Livny (1991). O2PL follows an optimistic read one, write all concurrency control approach, which allows lock processing for read locks on the site or the nearest server site. O2PL-MT then restricts the read unlock messages to remote server where the lock is set, by allowing unlocks to occur at the local or nearest server site.

The Bayou storage system as discussed by Terry, Theimer, Petersen, Demers, Spreitzer and Hauser (1995), provides a weakly consistent system that may ensure eventual consistency between the replicas. The Bayou architecture achieves high availability of data to mobile applications at the cost of providing weak consistency by allowing any client or server to perform updates (Demers *et al*, 1994). Conflict detection and resolution for the Bayou system are possible through the use of dependency checks and merge procedures, but is mainly application specific. That is, each write operation is attached with additional queries that check the server's data against an expected result, to specify if a conflict has occurred. A further merge procedure is included, in an attempt to resolve any conflict.

Pitoura and Bhargava (1995) and Pitoura (1996), propose a two-level consistency model in which clusters are formed from different partitions of a database system. Within each cluster, full mutual consistency is maintained while for data outside the cluster, only varying degrees of consistency are kept. This is possible through the introduction of strict and weak operations, where strict operations are the normal operations while the weak operations provides solutions that may be weakly consistent. Assuming, that the partitions are allocated through their importance on a cluster, then allowing varying consistencies between clusters is acceptable since the main users for any particular data would be within its cluster.

6.4 Transaction Support

Transaction support relies on all the necessary information to complete the operations being available. If data are unavailable the transaction would either fail or be required to wait for the data to be available. This may then result in transactions having long execution times.

Traditional transactions are a sequence of read and write operations, which may include operations accessing different data over multiple locations if a distributed database is used. Mobile transactions, however, are different and may display different characteristics to centralised and distributed systems. These characteristics may include the need to split a transaction into parts for computation on both local and remote databases, migration of the transaction to a remote non-mobile database when no user interactions are required to conserve the mobile computer's resources, support for the transaction by non-mobile databases, computation of different parts of a transaction on different servers while the mobile computer is on the move, and a long-lived nature due to the frequent disconnections that may occur (Madria, 1998; Dunham and Kumar, 1999).

A semantic-based transaction processing method has been extended for mobile transactions in Walborn and Chrysanthis (1995) who introduce the idea of fragmentable and reorderable objects to increase the concurrency to the entire database and improve the efficiency of caching. Using this idea, a mobile system may request a locally cached fragment on which transactions may operate. This fragment is consistent in that it is only accessible to the transactions of the mobile computer that requested it. Once the fragments are no longer required, they are merged back into the master database. Such a technique is suitable when data objects may be fragmented like aggregate items, sets and stacks.

Kangaroo Transactions were introduced by Dunham, Helal and Balakrishnan (1997) to allow transactions to be processed by capturing the moving nature of mobile computer. In effect a transaction is split up into several parts where each part is processed (and perhaps committed) on the server in which the mobile computer has moved. These parts or subtransactions are serialisable using the split transaction method as described by Pu, Kaiser and Hutchinson (1988) and may commit and abort independently.

Chrysanthis proposes an open-nested transaction model using the notion of Reporting Transactions and Co-Transactions for mobile computers to perform updates (Chrysanthis, 1993). Similarly, a mobile transaction may be split into several subtransactions that may commit and abort independently. A reporting transaction allows partial results to be shared with the rest of the subtransactions in the mobile transaction, while co-transactions allow the control of the partial results to be passed from one subtransaction to another at the time of sharing. Madria, Baseer and Bhowmick (2002) introduced a multi-version transaction model for mobile computing. The model aims to increase availability of data by using versions. By allowing a version of a transaction to be executed and committed on the mobile computer, the new data from that transaction would be readily available. A terminating version is then required at the main database to ensure the consistency of the database.

For multidatabase systems with mobile components, Yeo and Zaslavsky (1994) discussed a method for allowing multidatabase transactions to be submitted by a mobile computer. Once submitted, a mobile computer may disconnect and a coordinating site will process the transaction on its behalf. Using simple messaging and queuing techniques, the mobile computer may be informed of the status of the transaction when it reconnects. Such a technique is similar to having a proxy for the mobile computer that may process any transaction requested by the mobile computer.

6.5 System Recovery

In centralised and primary copy distributed database systems, the failure of the primary copy causes clients to be unable to effect an update. In instances where clients do not cache any kind of data locally, read accesses would also be denied. In cases where the failure is long in duration, a new site must be nominated to fill in the role as the primary copy database. The most common algorithm for deciding is an election algorithm as discussed by Garcia-Molina (1982). The algorithm specifies that a site nominate itself as primary and inform the remaining sites. Once a majority agreement is reached, the site is then the new coordinator until the actual primary coordinator has recovered. Similar algorithms may be used in mobile distributed systems with fixed sites where a fixed site may assume the role of primary coordinator. It is unlikely for a mobile computer to take on the role of primary coordinator since they may disconnect frequently from the network.

In mobile *ad-hoc* networks, network partitioning occurs frequently when mobile computers move from one area to another. Network partitioning occurs when a network is split into different partitions in such a way that communication between partitioned networks is not possible. Aggarwal *et al* introduced a clustering algorithm, where a super master is elected that will know all the mobile computers that are available, to design and create a scatternet (Aggarwal, Kapoor, Ramachandran and Sarkar, 2000).

7. CONCLUSION

Mobile distributed databases suffer from additional issues that are not usual in traditional or distributed databases. The dependency of mobile computers on batteries can exacerbate poor network connectivity and a reduction in available storage capacity. Data summarisation techniques may be an important tool to provide better availability of data and more efficient response to queries in the mobile environment.

This paper examined several techniques currently proposed for the mobile environment. It also discussed the different database functionality required by mobile distributed databases and how different research efforts have proposed to deal with them. While the most common techniques for summarisation in mobile databases are based only on the reduction in volume, it is also important to examine the context of the users that will use the mobile computer. We would argue, therefore, that a context-sensitive summarisation technique would provide more efficient allocation of data to provide more data availability and better response to queries.

REFERENCES

AGGARWAL, A., KAPOOR, M., RAMACHANDRAN, L. and SARKAR, A. (2000): Clustering algorithms for wireless ad hoc networks, *4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Boston, MA, 54–63.

AHMAD, I., KARLAPALEM, K., KWOK, Y.-K. and SO, S.-K. (2002): Evolutionary algorithms for allocating data in distributed database systems, *Distributed and Parallel Databases* 11(1): 5–32.

BARBARA, D., DUMOUCHEL, W., FALOUTSOS, C., HAAS, P. J., HELLERSTEIN, J.M., IOANNIDIS, Y.E., JAGADISH, H.V., JOHNSON, T., NG, R., POOSALA, V., ROSS, K.A. and SEVCIK, K.C. (1997): The New Jersey data reduction report, *IEEE Data Engineering Bulletin: Special Issue on Data Reduction Techniques* 20(4): 3–45.

BELL, T., WITTEN, I.H. and CLEARY, J.G. (1989): Modeling for text compression, *ACM Computing Surveys* (CSUR) 21(4): 557–591.

BERNSTEIN, P.A., HADZILACOS, V. and GOODMAN, N. (1987): Concurrency control and recovery in database systems, Addison-Wesley.

BREITBART, Y., KOMONDOOR, R., RASTOGI, R., SESHADRI, S. and SILBERSCHATZ, A. (1999): Update propagation protocols for replicated databases, in *ACM Int. Conf. on Management of Data (SIGMOD)*, Philadelphia, 97–108.

BRUNSTROM, A., LEUTENEGGER, S.T. and SIMHA, R. (1995): Experimental evaluation of dynamic data allocation strategies in a distributed database with changing workloads, in *4th International Conf. on Information and Knowledge Management (CKIM)*.

CANNANE, A. and WILLIAMS, H.E. (2001): General-purpose compression for efficient retrieval, *Journal of the American Society for Information Science and Technology* 52(5): 430–437.

CANNANE, A., WILLIAMS, H.E. and ZOBEL, J. (1999): A general-purpose compression scheme for databases, *Data Compression Conference*, 519.

CAREY, M.J. and LIVNY, M. (1991): Conflict detection tradeoffs for replicated data, *ACM Transactions on Database Systems* 16(4): 703–746.

CERI, S., NEGRI, M. and PELAGATTI, G. (1982): Horizontal data partitioning in database design, *Proc. 1982 ACM SIGMOD international conference on Management of data*, ACM Press, Orlando, Florida, 128–136.

CHAN, D. and RODDICK, J.F. (2003): Context-sensitive mobile database summarisation, OUDSHOORN, M. ed., 26th Australasian Computer Science Conference (ACSC2003), Vol. 16 of Conferences in Research and Practice in Information Technology, ACS, Adelaide, Australia, 139–149.

CHRYSANTHIS, P.K. (1993): Transaction processing in a mobile computing environment, *Proc. IEEE workshop on Advances in Parallel Distributed System*, 77–82.

CHU, P.-C. (1992): A transaction-oriented approach to attribute partitioning, *Information Systems* 17(4), 329–342.

CORMACK, G.V. (1985): Data compression on a database system, *Communications of the ACM* 28(12): 1336–1342.

CORNELL, D.W. and YU, P.S. (1990): A vertical partitioning algorithm for relational databases, *Third International Conference on Data Engineering, IEEE*, 30–35.

COVINGTON, M.J., LONG, W., SRINIVASAN, S., DEV, A.K., AHAMAD, M. and ABOWD, G.D. (2001): Securing context-aware applications using environment roles, in *Proc. Sixth ACM symposium on Access control models and technologies*, ACM Press, Chantilly, Virginia, US, 10–20.

DAVIDSON, S.B., GARCIA-MOLINA, H. and SKEEN, D. (1985): Consistency in partitioned network’, *ACM Computer Surveys* 17(3), 341–370.

DEMERS, A.J., PETERSEN, K., SPREITZER, M.J., TERRY, D.B., THEIMER, M.M. and WELCH, B.B. (1994): The Bayou architecture: Support for data sharing among mobile users, *Proc. IEEE Workshop on Mobile Computing Systems & Applications*, Santa Cruz, California, 2–7.

DONG, G. and MOHANIA, M. (1996): Algorithms for view maintenance in mobile databases, *First Australian Workshop on Mobile Computing and Databases*, Monash University, Australia.

DOUGLIS, F., KAASHOEK, M.F., LI, K., CÁCERES, R., MARSH, B. and TAUBER, J. (1994): Storage alternatives for mobile computers, *First Symposium on Operating Systems Design and Implementation*, Monterey, CA, USA, 25–37.

DUNHAM, M.H. and HELAL, A. (1995): Mobile computing and databases: Anything new?, *SIGMOD Record* 24(4): 5–9.

DUNHAM, M.H., HELAL, A. and BALAKRISHNAN, S. (1997): A mobile transaction model that captures both the data and movement behavior, *Mobile Networks and Applications* 2(2): 149–162.

DUNHAM, M.H. and Kumar, V. (1999): Impact of mobility on transaction management, *Proc. International Workshop on Data Engineering for Wireless and Mobile Access*, Seattle, WA, USA, 14–21.

ELMASRI, R. and NAVATHE, S. (2000): Fundamentals of database systems, 3rd edn, Addison-Wesley, Redwood City, CA.

FIFE, L.D. and GRUENWALD, L. (2003): Research issues for data communication in mobile ad-hoc network database systems, *SIGMOD Record* 32(2): 42–47.

GANGULY, S. and ALONSO, R. (1993): Query optimization in mobile environments, *Fifth Workshop on Foundations of Models and Languages for Data and Objects*, Germany, 1–17.

GARCIA-MOLINA, H. (1982): Elections in a distributed computing system, *IEEE Transactions on Computers* 31(1): 48–59.

GRAEFE, G. and SHAPIRO, L.D. (1991): Data compression and database performance, *Proc.ACM/IEEE-CS Symposium on Applied Computing*, Kansas City, MO.

GRAY, J., HELLAND, P., O'NEIL, P. and SHASHA, D. (1996): The dangers of replication and a solution, *1996 ACM SIGMOD International Conference on Management of Data*, 173–182.

HALEVY, A.Y. (2001): Answering queries using views: A survey, *The VLDB Journal* 10(4): 270–294.

HAN, J. and FU, Y. (1994): Dynamic generation and refinement of concept hierarchies for knowledge discovery in databases, *KDD Workshop*, 157– 168.

HAWICK, K. and JAMES, H.A. (2003): Middleware for context sensitive mobile applications, *Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing*, Adelaide, Australia.

HEUER, A. and LUBINSKI, A. (1996): Database access in mobile environments, *Database and Expert Systems Applications*, 544–553.

HEUER, A. and LUBINSKI, A. (1998): Data reduction – an adaptation technique for mobile environments, *In Interactive Applications of mobile Computing (IMC'98)*.

HOLLIDAY, J., AGRAWAL, D. and ABBADI, A.E. (2000): Planned disconnections for mobile databases, *DEXA Workshop*, 165–172.

HUANG, Y., SISTLA, P. and WOLFSON, O. (1994): Data replication for mobile computers, *1994 ACM SIGMOD*, 13–24.

IMIELINSKI, T. and BADRINATH, B.R. (1994): Mobile wireless computing: Challenges in data management, *Communications of the ACM* 37(10): 18–28.

JING, J., BUKHRES, O. and ELMAGARMID, A. (1995): Distributed lock management for mobile transactions, *Proc.of the 15th International Conference on Distributed Computing Systems*, Vancouver, Canada, 118–125.

JONES, G.J. and BROWN, P.J. (2004): Context-aware retrieval for ubiquitous computing environment, *Mobile and Ubiquitous Info Access Ws 2003*, 227–243.

KEUNNING, G.H. and POPEK, G.J. (1997): Automated hoarding for mobile computers, *Proc. Sixteenth ACM Symposium on Operating Systems Principles*, ACM Press, Saint Malo, France, 264–275.

LADIN, R., LISKOV, B., SHRIRA, L. and GHEMAWAT, S. (1992): Providing high availability using lazy replication, *ACM Transactions on Computer Systems* 10(4): 360–391.

LAUZAC, S.W. and CHRYSANTHIS, P.K. (1998a): Programming views for mobile database clients, *DEXA Workshop*, 408–413.

LAUZAC, S.W. and CHRYSANTHIS, P.K. (1998b): Utilizing versions of views within a mobile environment, *The 9th ICCI*.

LAW, C. and Siu, K.-Y. (2001): A bluetooth scatternet formation algorithm, *Proc. IEEE Symposium on Ad Hoc Wireless Networks*.

LELEWER, D.A. and HIRSCHBERG, D.S. (1987): Data compression, *ACM Computing Surveys (CSUR)* 19(3): 261–296.

LEOPOLD, R.J., MILLER, A. and GRUBB, J.L. (1993): Iridium system: A new paradigm in personal communications, *Applied Microwave and Wireless* 5(4): 68–78.

LUBINSKI, A. (2000): Small database answers for small mobile resources, *International Conference on Intelligent Interactive Assistance and Mobile Multimedia Computing (IMC2000)*, Rostock, November, 9-10.

LUBINSKI, A. and HEUER, A. (2000): Configured replication for mobile applications, *Baltic DB&IS*'2000, Vilnius, Lithuania.

MADRIA, S.K. (1998): Transaction models for mobile computing, *International Database Engineering and Application Symposium*, Singapore, 92–102.

MADRIA, S.K., BASEER, M. and BHOWMICK, S.S. (2002): A multi-version transaction model to improve data availability in mobile computing, *CoopIS/DOA/ODBASE 2002*, 322–338.

MADRIA, S.K., MOHANIA, M.K. and RODDICK, J.F. (1998): A query processing model for mobile computing using concept hierarchies and summary databases, *Foundations of Database Organisation*, 146–157.

MUTHURAJ, J., CHAKRAVARTHY, S., VARADARAJAN, R. and NAVATHE, S.B. (1993): A formal approach to the vertical partitioning problem in distributed database design, *Proc.of Second International Conference on Parallel and Distributed Information Systems*, San Diego, California.

NAVATHE, S.B., CERI, S., WIEDERHOLD, G. and DOU, J. (1984): Vertical partitioning algorithms for database design, *ACM Transactions on Database Systems* 9(4): 680–710.

NAVATHE, S.B., KARLAPALEM, K. and RA, M. (1996): A mixed fragmentation methodology for initial distributed database desig, *Journal of Computer and Software Engineering* 3(4).

NAVATHE, S.B. and RA, M. (1989): Vertical partitioning for database design: A graphical algorithm, *ACM SIGMOD International Conference on Management of Data*, Vol. 18, Portland, 440–450.

NEUMANN, K. and MASKARINEC, M. (1997): Mobile computing within a distributed deductive database, *Proc. 1997 ACM symposium on Applied computing*, ACM Press, San Jose, California, United States, 318– 322.

PHATAK, S.H. and BADRINATH, B.R. (1999): Data partitioning for disconnected client server databases, *MobiDE*, 102–109.

PITOURA, E. (1996): A replication schema to support weak connectivity in mobile information systems, *7th International Conference on Database and Expert Systems Applications (DEXA)*.

PITOURA, E. and BHARGAVA, B. (1993): Dealing with mobility: Issues and research challenges, Technical report, Department of Computer Science, Purdue University, US.

PITOURA, E. and BHARGAVA, B. (1995): Maintaining consistency of data in mobile distributed environments, *15th International Conference on Distributing Computing Systems*, Vancouver, British Columbia, Canada, 404– 413.

PITOURA, E. and SAMARAS, G. (1998): Data management for mobile computing, Kluwer Academic Publishers.

PU, C., KAISER, G.E. and HUTCHINSON, N. (1988): Split-transactions for open-ended activities, *Proc. 14th Conference on Very Large Databases*, Los Altos, CA.

RAY, G., HARITSA, J.R. and SESHADRI, S. (1995): Database compression: A performance enhancement tool, *International Conference on Management of Data*, Pune, India.

REITER, R. (1978): On closed world databases, in GALLAIRE, H. and MINKER, J. eds, Logic and Databases, Plenum Press, New York, 55–76. Reprinted in Artificial Intelligence and Databases. MYLOPOULOS, J. and BRODIE, M.L. (eds.), Morgan Kaufmann, 248–258.

RODDICK, J.F. (1997): The use of overcomplete logics in summary data management, *8th Australasian Conference on Information Systems*, Adelaide, Australia.

RODDICK, J.F., MOHANIA, M.K. and MADRIA, S.K. (1999): Methods and interpretation of database summarisation, *Database and Expert Systems Applications*, 604–615.

ROTH, M.A. and VAN HORN, S.J. (1993): Database compressio, *ACM Sigmod Record* 22(3): 31–39.

SEVERANCE, D.G. (1983): A practitioner's guide to data base compression, *Information Systems* 8(1): 51–62.

SHEPHERD, J., HARANGSRI, B., CHEN, H.L. and NGU, A. (1995): A two-phase approach to data allocation in distributed databases, *Database Systems for Advanced Applications*, 380–387.

STANOI, I., AGRAWAL, D., EL ABBADI, A., PHATAK, S.H. and BADRINATH, B.R. (1999): Data warehousing alternatives for mobile environments, *MobiDE*, 110–115.

TERRY, D.B., THEIMER, M.M., PETERSEN, K., DEMERS, A.J., SPREITZER, M.J. and HAUSER, C.H. (1995): Managing update conflicts in bayou, a weakly connected replicated storage system, *15th ACM Symposium on Operating Systems Principles*, ACM, Copper Mountain Resort, Colorado.

WALBORN, G.D. and CHRYSANTHIS, P.K. (1995): Supporting semantics-based transaction processing in mobile database applications, *14th IEEE Symposium on Reliable Distributed Systems*.

WIESMANN, M., PEDONE, F., SCHIPER, A., KEMME, B. and ALONSO, G. (2000a): Database replication techniques: a three parameter classification, *Proc.19th IEEE Symposium on Reliable Distributed Systems*, IEEE Computer Society, Nürenberg, Germany.

WIESMANN, M., PEDONE, F., SCHIPER, A., KEMME, B. and ALONSO, G. (2000b): Understanding replication in databases and distributed systems, *Proc. 20th International Conference on Distributed Computing Systems ICDCS'2000*, IEEE Computer Society Technical Committee on Distributed Processing, Taipei, Taiwan, R.O.C., 264–274.

WOLFSON, O. and JAJODIA, S. (1995): An algorithm for dynamic data distribution in distributed systems, *Information Processing Letters* 53: 113–119.

YEO, L.H. and ZASLAVSKY, A.B. (1994); Submission of transactions from mobile workstations in a cooperative multidatabase processing environment, *Proc.14th IEEE International Conference on Distributed Computing*, 372–379.

ZASLAVSKY, A.B. and TARI, Z. (1998): Mobile computing: Overview and current status, *Australian Computer Journal* 30(2): 42–52.

ZENEL, B. and DUCHAMP, D. (1995): Intelligent communication filtering for limited bandwidth environments, Colombia University, in *IEE Fifth Workshop on Hot Topics in Operating System*, Rosario WA.

BIOGRAPHICAL NOTES

Darin Chan is a PhD student in the School of Informatics and Engineering, Flinders University. He holds a Bachelor of Engineering (IT&T) with Honours from the University of Adelaide. His recent research interests include mobile databases, summarisation, relational and distributed databases.

Darin Chan

Professor John Roddick is currently the SACITT Chair in Information Technology and Associate Head (Research) in the School of Informatics and Engineering at Flinders University. He joined Flinders in 2000 after 10 years at the University of South Australia and five years at the University of Tasmania. This followed 10 years experience in the computing industry as (progressively) a programmer, analyst, project leader and consultant. Professor Roddick has published over 100 papers in a number of areas of computing but specialises in the fields of Data Mining and Knowledge Discovery – specifically in temporal and spatial data mining and as applied to medical and health data, and Conceptual Modelling – specifically in enhanced database systems semantics such as schema evolution and temporal and spatial systems design and use. He holds a PhD from La Trobe University, an MSc from Deakin University and a BSc(Eng)(Hons) from Imperial College, London.

John Roddick