Mobile Databases: a Selection of Open Issues and Research Directions *

Action members **

[1]EPFL, [2]U. Grenoble, [3]INRIA-Nancy, [4]INT-Evry, [5]U. Montpellier 2, [6]U. Paris 6, [7]U. Versailles

Abstract

This paper reports on the main results of a specific action on mobile databases conducted by CNRS in France from October 2001 to December 2002. The objective of this action was to review the state of progress in mobile databases and identify major research directions for the French database community. Rather than provide a survey of all important issues in mobile databases, this paper gives an outline of the directions in which the action participants are now engaged, namely: copy synchronization in disconnected computing, mobile transactions, database embedded in ultra-light devices, data confidentiality, P2P dissemination models and middleware adaptability.

1 Introduction

At the end of 2001, the National Center for Scientific Research (CNRS) in France launched a number of specific actions (AS) in order to identify the most challenging issues to be investigated by the French research community (http://www.cnrs.fr/STIC/actions/as/as.htm). The impact of ubiquitous and pervasive computing in many aspects of our everyday life motivated the CNRS to fund a specific action on mobile databases, in October 2001 for an initial period of one year. This paper reports on the research directions selected by the action participants.

Different classes of mobile applications can be distinguished depending on the data management requirements they introduce. The most common applications today are Mobile Client – Fixed Host. Examples involve travelling employees accessing a fixed corporate database, mobile users accessing personal data (e.g., banking data, agenda, bookmarks) hosted by a Data Service Provider (DSP) or servers broadcasting information (e.g., traffic, weather, stock exchange) towards a large population of users [CFZ01]. Specific requirements concern data consistency (while the primary copy of the data is managed by the server, replica can be temporarily hosted by mobile devices to allow disconnected computing), data confidentiality (as data can be hosted by an untrusted DSP) and data availability (new data access models have to be devised in this context).

Portable folders (e.g., medical folder on a smart card, phonebook on a SIM card) are good representative of the Mobile Client – Mobile Host class of applications [PBVB01]. The requirements concern the management of database embedded in ultra-light devices, data durability (replicas have to be managed on the fixed network to enforce crash resiliency) and copy synchronization in the case of collaborative applications (e.g., to fix a meeting date among agendas hosted by PDA). More generally, Mobile Host applications will gain popularity with the emergence of ambient intelligence, whatever be the client mobility. For example, the vision of the future dataspace paints a physical space enhanced with digital information made available through large-scale ad-hoc sensor networks [ImN02]. This introduces new requirements in terms of query execution to access an information scattered on a widely distributed population of weakly connected ultra-light devices.

Even Fixed Client – Fixed Host applications may involve mobile databases. Much work has been conducted on moving objects [WXC98], e.g., to keep track of mobile locations in telecommunication, transportation or traffic regulation applications. However, we do not cover this issue in this paper.

Achieving these requirements faces strong hardware and software constraints introduced by the computing environment: low bandwidth of wireless communication channels, frequent disconnections, scarce computing, storage and energy resources of the mobile devices and inadequacy of the middleware initially designed to handle fixed and wired clients and servers.

The remainder of this paper is organized as follows. Sections 2 to 7 are each devoted to one issue addressed by one or more participants in this specific action, namely: copy synchronization in disconnected computing, mobile transactions, database embedded in ultra-light devices, confidentiality of data hosted by untrusted DSP, P2P dissemination models and middleware adaptability to mobility and disconnection. Section 8 concludes.

* This research action, headed by Philippe Pucheral, was funded by CNRS, the National Center for Scientific Research in France.

** Guy Bernard[4], Jalel Ben-Othman[7], Luc Bouganim[7], Gérôme Canals[3], Sophie Chabridon[4], Bruno Defude[4], Jean Ferrié[5], Stéphane Gançarski[6], Rachid Guerraoui[1], Pascal Molli[3], Philippe Pucheral[7], Claudia Roncancio[2], Patricia Serrano-Alvarado[2] and Patrick Valduriez[6].

2 Data synchronization

Disconnected work is an important issue for supporting mobility. Disconnected work can be achieved by replicating data on the mobile device before disconnection and by merging divergent data when connection is again available. This can occur with a single user and several devices. In this case, all copies are managed by the same user but at different times and places. Disconnected work can also occur with multi-synchronous groupware. Each member manages her copies and propagates her updates to others when connection is available. Different tools allow to synchronize divergent data. File Synchronizers (Microsoft's Briefcase, Power Merge, Windows File Synchronizer, ...) propagate non-conflicting updates from one replica to another and delegate conflict resolution to users [BaP98]. But how to characterize a conflict and what happens to conflicting updates? Data Synchronizers (Palm Pilot HotSync, Puma Technology Intellisync, Microsoft ActiveSync, Apple I-Sync ...) go one step further and allow synchronizing the file contents. This requires tools to be aware of data types but the synchronization correctness is not formally defined. As File Synchronizers, these tools just propagate non-conflicting updates to other replicas and delegate conflict resolution to users. In configuration management, merge tools (Diff3, rcsMerge, XMLDiffMerge) are used to synchronize the file system and some kind of files (text, XML,). The existence of so many tools doing a similar task raises important issues: Is there a generic synchronizer? Is there a theory of synchronization? What is a correct synchronization?

An interesting approach has been developed in the domain of Real-time Groupware [SuE98, VCF00]. The transformational approach uses the semantics of operations applied to copies to enforce three properties, namely: (i) causality preservation, (ii) intention preservation and (iii) convergence. These properties are used as a correctness criterion for synchronizing data. The transformational approach considers *n* sites. Each site hosts a local copy of the shared objects. When an object is modified at one site, the operation is executed immediately and sent to the other sites to be executed remotely again. Thus, every operation is processed in four steps: (1) generated at one site, (2) broadcasted to the other sites, (3) received by them and (4) executed remotely on them. The execution context of a received operation op_i may be different from its generation context. In this case, the integration of op_i by remote sites may lead to inconsistencies between replicas. Received operations have to be transformed according to local concurrent operations before their re-execution. This is done by using Operational Transformations [SuE98]. The Operational Transformations model defines two main components: the integration algorithm and the transformation functions. The Integration algorithm is responsible for receiving, broadcasting and executing operations. It is independent of the type of shared data and uses transformation functions when needed. The transformation functions are responsible for merging two concurrent operations defined on the same state. They are specific to the type of shared data.

Our objective is to build a safe generic synchronizer based on the transformational model enforcing the properties of convergence, causality and intention preservation. The adaptability of the synchronizer relies on the definition of data specific transformation functions. We distinguish two research directions. First, the transformational model has been primarily designed to support short periods of disconnection. It must be extended to tolerate longer disconnections [IMO03]. Second, it is common to move a log of operations from or to a mobile device. Thus, it is important to minimize the size of the log. Backward transformations [SCF98] can be used to compress the log.

3 Mobile transactions

In a mobile environment, transactions may be initiated by mobile units (MU) and distributed among a set of mobile or fixed units. Mobile transactions raise at least two specific problems [SRA01]: MU may move during transaction execution and they may be partially/totally disconnected. MU movements require specific procedures to support data access from the MU through the closest base station (e.g. Kangaroo, MDSTPM). MU disconnections have an impact on transaction coherency. Indeed, disconnected work implies that local copies must be installed at the MU. Local transactions can be locally committed but effective durability requires a connection for global commit. Copy management (creation, reintegration) has to be handled by the transaction manager. Furthermore, the long, unbounded disconnected periods have important consequences on transaction isolation. On one hand, the use of pessimistic approaches can make data unavailable for a too long – potentially infinite – time. On the other hand, optimistic mechanisms may lead to important divergence between the server and local copies, making their reintegration difficult. This problem is partially solved by either using time-outs to mix optimistic and pessimistic CC and partitioning data [MoV00], or by providing mechanisms to ease the reintegration of local copies [PiB99,WaC99]. Finally, disconnections raise a new problem on the global commit phase: the commit protocol should not be blocking (as 2PC is). New solutions must be used, such as unilateral commit or time-out based [Kum00].

To go further, we propose to develop a new framework for mobile transactions. Transactions at MU are grouped into long-duration transactions (LDT) with three phases. First, the local environment (including local copies) is initialized at the MU in connected mode. Then LDT are performed at the MU, possibly in disconnected mode, and are locally committed. Finally, LDT are globally committed and local copies are reintegrated if possible, when the MU gets a stable connection with the fixed server. The main idea is to associate each LDT with a *contract*. Intuitively, a contract can be seen as a set of rules that guarantee a successful global commit if satisfied. This idea generalizes the notions found in [WaC99]. A contract defines each partner engagement regarding isolation: access intentions, locks, timeouts for locks, divergence limit for shared copies, etc. It is established during the initialization phase but can be renegociated during the execution phase, when temporary connections can be established. Contracts may also be used to guide the reintegration process in the final phase of the LDTs.

The main issues we plan to investigate are related to the notion of contract: how to specify them, how to derive operational rules to maintain them at execution time and how to use them to facilitate the reintegration of locally modified copies with the server copies in order to increase the global commit rate.

4 Embedded databases

Pervasive computing is now a reality and intelligent devices flood many aspects of our everyday life. As new applications appear, the need for database techniques embedded in various forms of lightweight computing devices arises. Personal folders on chips, networks of sensors and data hosted by autonomous mobile computers are different illustrations of the need for evaluating queries confined in hardware constrained computing devices.

Sensor networks gathering weather, pollution or traffic information have motivated several recent works. The COUGAR project at Cornell [BGS01] defined the concept of "Device Database System"; a database system that enables distributed query processing over a sensor network. Device Database Systems have brought out the need for executing local computation on the data, like aggregation [MaF02], sort and top-n queries, either to save communication (impacting significantly the device power consumption) in push-based systems or to participate in distributed pull-based queries [MFH02].

Hand-held devices are other forms of autonomous mobile hosts that can be used to execute on-board queries. Light versions of popular DBMS like Sybase Adaptive Server Anywhere, Oracle 8i Lite, SQLServer for Windows CE or DB2 Everyplace have been designed for such devices. The database software vendors mainly focused on deployment, data synchronization issues and DBMS footprint reduction (by simplifying and componentizing the DBMS code).

Personal folders on chip constitute another motivation to execute on-board queries. Typically, smart cards are used in various applications involving personal data (such as healthcare, insurance, phone books etc.). In this context, queries can be fairly complex and their execution must be confined on the chip to prevent any disclosure of confidential data.

The resource asymmetry of lightweight devices entails a thorough re-thinking of the database techniques. Our objective is to design embedded database components that can match highly constrained hardware resources. Preliminary studies led us to design and validate a full-fledged DBMS, called PicoDBMS [PBV01], embedded in an advanced smart card platform (powerful CPU, tiny RAM, fast read time and dramatically slow write time in EEPROM stable storage). PicoDBMS is based on a compact and efficient ad-hoc storage and indexing model. The PicoDBMS query processor takes advantage of the CPU power and the fast read time of the EEPROM to process any type of SQL query without using RAM, whatever be the volume of the queried data.

Capitalizing on this work, new important research efforts have to be undertaken: (i) to better capture the impact of each device hardware constraint on database techniques, (ii) to propose new storage, indexing and query techniques allowing to build ad-hoc embedded database components and (iii) to set up co-design rules to help calibrating the hardware resources of future devices to match specific applications' requirements.

As a first step in this research direction, we made a thorough analysis of the RAM consumption problem [ABP03] in the absence of index. We try to answer three important questions: (i) does a memory lower bound exist whatever be the volume of the queried data? (ii) how can a query be optimized without hurting this lower bound? (iii) how an incremental growth of RAM impacts the techniques devised in a lower bound context. This study provides helpful guidelines to calibrate the RAM resource of a hardware platform according to given application's requirements as well as to adapt an application to an existing hardware platform. We are now studying the combination of our proposed techniques with indices. Next, we plan to focus on energy consumption and on the usage of future stable memory technologies.

5 Data privacy and confidentiality

The democratization of ubiquitous computing (access data anywhere, anytime, anyhow) strongly emphasizes the need for data security. Typically, the increasing connection of travelling users to corporate databases and the resort to Web hosting companies and DSP to make personal data available to mobile users introduce new threats on data privacy and confidentiality. Users have no other choice than trusting Web companies' privacy policies. However, database attacks are more and more frequent (their cost is estimated to more than $100 billion per year) and 45% of the attacks are conducted by insiders [FBI02]. Notorious examples of privacy policy violations are given in [AKL02] to motivate the design of Hippocratic DBMS, that is, DBMS able to take a confidentiality oath.

While traditional database security principles, like user authentication, communication encryption and server-enforced access controls [BPS96] are widely accepted, they remain inoperative against insider attacks. Several attempts have been made to strengthen server-based security approaches with database encryption [Ora02]. However, as Oracle confesses, server encryption is not the expected "armor plating" because the Database Administrator (or an intruder usurping her identity) has enough privilege to tamper the encryption mechanism and get the clear-text data.

Client-based security approaches have been recently investigated. They still rely on database encryption, but encryption and decryption occur only at the client side to prevent any disclosure of clear-text data at the server. Storage Service Providers proposing encrypted backups for personal data are crude representative of the client-based security approach. The management of SQL queries over encrypted data complements well this approach [HIL02]. These solutions provide a convincing way to store and query safely personal data on untrusted servers. However, sharing data among several users is not addressed. Actually, users willing to share data have to share the same encryption keys, thus inheriting the same access rights on the data.

Our main research objective is to address this sharing issue. We are working on a solution called C-SDA (Chip-Secured Data Access), which allows querying encrypted data while controlling personal privileges [BoP02]. C-SDA is a client-based security component acting as an incorruptible mediator between a client (potentially mobile) and an encrypted database. This component is embedded into a smart card to prevent any tampering to occur on the client side. This cooperation of hardware and software security components constitutes a strong guarantee against attacks and allows re-establishing the orthogonality between access right management and data encryption. This work is partly supported by ANVAR, the national agency for the promotion of research in France, and Schlumberger who provides us with advanced smart card platforms. This initial work considers relational data, pull-based queries and an SQL compliant access right model. New solutions need be devised to tackle more complex data (e.g., semi-structured data like XML), push-based models (selective information broadcasting) and sophisticated access right models (roles, organisations, permissions, prohibitions, obligations …). The recent advances in secured computing platforms (secured co-processors, smart tokens, Trusted Computed Platform Alliance) deserve a particular attention by allowing more general, powerful and efficient solutions.

An alternative to the preceding approaches is to embed the user's confidential data into her own mobile device (e.g., a PDA). Apart from their limitation in terms of storage capacity, even these devices cannot be fully trusted because they can be stolen, lost or destroyed (thus a copy of the data they host has to be maintained in the network to guarantee data resiliency). This raises the issue of building secured portable folders, using the secured mobile devices which we discussed in Section 4.

6 P2P dissemination models

The heart of an information system has traditionally been the central database server. Several client processes can store and retrieve information of interest through this server, which centralizes all requests and updates. In a mobile environment, relying on a central server does hardly make any sense: the server easily becomes a single point of failure and a performance bottleneck. No computation can be achieved unless the connection to the central server is up and running. One might instead consider implementing the information system on a cluster of machines that provides acceptable performance and fault-tolerance. However this does not solve completely the scalability issue: first, as the number of client processes grows, the cluster of servers might become a performance bottleneck and, secondly, consistency between servers needs to be addressed and this is tricky with mobile clients accessing the database cluster.

A radically different approach is to decentralize the database maintenance activity, by distributing among every client process in the system the tasks of information replication and transmission. The very notion of server disappears. Or more accurately, every process ends up playing part of the role of the server, besides its client role. A process, acting as a client that wishes to disseminate a new piece of information to the system does not send it to a server, or a cluster of servers, in charge of forwarding it but rather to a set of other peer processes, chosen at random. In turn, each of these processes does the same, and also forwards the information to a randomly selected processes, and so forth.

One major difficulty with this approach lies in ensuring that any given information is eventually accessible to those clients with an interest in it. A client that is interested in some information might indeed access other clients, but these might not have the information. Of course, in a large-scale environment, one cannot guarantee that any information produced by some client is accessible to all clients, but a probabilistic guarantee might be enough.

This is what epidemic algorithms [DGH87] aim at providing. They are simple, easy to deploy, provide probabilistic guarantees of delivery and their performance degrade gracefully in the presence of network or computing nodes failures. The principle underlying information dissemination mimics the spread of epidemics (an even closer analogy is with the spread of a rumor among humans via gossiping), and we talk about epidemic information dissemination [BHO99, RBV03]. Several hard issues need be solved to provide scalable implementations of such algorithms so that they can be applied in practice in Internet-wide settings.

The questions to be addressed include: (1) how do processes get to know each other, and how many do they need to know (membership maintenance) [EGH03]? (2) How to make the connections between processes reflect the actual network topology such that the performance is acceptable (network awareness)? (3) which information to drop at a process when its storage buffer is full (buffer management)? (4) how to take into account the actual interest of processes and decrease the probability that they receive and store information of no interest to them (message filtering) [EuG02]?

7 Middleware adaptability

Nowadays, users become more and more nomadic. Moreover, nomadicity often goes with the use of a mobile terminal (such as a PDA or a smartphone) with wireless networking capability. This new context involves the need of adaptation at various levels (applications, middleware, system resources usage) for enabling users to access their favorite applications or data, from any place and any terminal type.

As stated in the introduction, the mobility context imposes strong hardware and software constraints. These constraints lead to a dilemma for architecturing distributed applications: for the one hand, rely on servers as much as possible (because of resource scarcity on the terminal); for the other hand, favor autonomous functioning (because of poor performance and cost of wireless links) [Sa96]. Moreover, the environment state fluctuates with both location and time, so the adaptation mechanism should be dynamic.

Pioneering work in the area of dynamic adaptation was the CMU Odyssey project. Odyssey considered the adaptation to current available bandwidth [NSN97] by introducing the notion of "fidelity" of multimedia or cartographic data – the corresponding files are stored in the servers in several versions with various precisions and storage size amounts. The adaptation is done at the terminal side only, by the means of a cooperation between the system (providing the monitoring and notification mechanism) and the application (implementing the programmer/user defined policy) for dynamically switching from one fidelity level to another according to the changes in the environment (bandwidth or battery). The MIT Rover project considered the adaptation to network disconnections. The Rover toolkit [JTK97] is made of two complementary mechanisms, namely the structuring of application code in terms of relocatable data objects (that can be dynamically transferred between server and terminal), and an asynchronous RPC protocol.

There are several research directions. First, compatibility with (de-jure or de-facto) standards – Odyssey relies on a modified Unix kernel, and the Rover object programming model is proprietary – is being considered in the Java/CORBA domain; to this end, we have developed a general framework (DOMInt, [CCB02]) for disconnection handling in the Java/CORBA domain, and we are extending this work to other domains (such as SOAP and Web Services). Second, the middleware layer, interposed between the applications and the operating system, appears to be the right place to implement the adaptation mechanisms and policies, especially with the reflection and self-adaptability properties that are arising [MCE03]. The approach taken in Nexus [FKV00], currently restricted to location-awareness, could be extended to other dimensions of mobility. We are investigating how handling non-functional properties (such as bandwidth variability) can be done through reflection properties. Third, the concept of "fidelity" could be extended to cover dimensions such as temporal consistency, semantic consistency, security level, or data precision. The challenge here is to find a way to describe a multi-dimensional fidelity level, and to translate applications' requirements into resource parameters. We are extending user profile expression languages to this end. Finally, the concept of multi-dimensional data fidelity appears promising for driving "data recharging" during connection periods of wireless mobile terminals [Ch01].

8 Conclusion

In this paper, we reviewed the state of progress in mobile databases and identify some major research directions for the French database community. By analyzing the data management requirements of mobile applications under the strong hardware and software constraints introduced by the computing environment, we identified a number of research directions: copy synchronization in disconnected computing, mobile transactions, database embedded in ultra-light devices, data confidentiality, P2P dissemination models and middleware adaptability. For each research direction, we defined the problem, we reviewed existing solutions and we outlined our research agenda.

Our analysis of the various dimensions of the problem illustrates the richness and the grand challenge of the field of mobile databases from a research point of view. Each dimension offers its own research directions. Beyond the improvement of existing solutions in terms of performance, scale-up, quality, adaptability, etc., we can note the needs for more generality of the solutions, typically restricted to one application domain, for formal studies of the proposed models and languages, and for validation and experimentation in real mobile settings. Finally, the constant introduction of new applications that strive to take advantage of new mobile computing capabilities will keep feeding the research agenda of mobile databases for a long time.

9 References*

[ABP03] Anciaux N., Bouganim L., Pucheral P., "Memory Requirements for Query Execution in Highly Constrained Devices", Int. Conf. on Very Large Data Bases (VLDB), 2003.

[AKS02] Agrawal R., Kiernan J., Srikant R., Xu Y., "Hippocratic Databases", Int. Conf. on Very Large Data Bases (VLDB), 2002.

* Due to space limitation, the reference list is not intended to be a complete list on all issues discussed in the paper.

[BGS01] Bonnet P., Gehrke J., Seshadri P., "Towards Sensor Database Systems". Int. Conf. on Mobile Data Management, 2001.

[BHO99] Birman K.P., Hayden M., Ozkasap O., Xiao Z., Budiu M., Minsky Y.., "Bimodal multicast". ACM Transactions on Computer Systems, 17(2), 1999.

[BaP98] Balasubramaniam S., Pierce B., "What is a File Synchroniser", ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom), 1998.

[BoP02] Bouganim L., Pucheral P., "Chip-Secured Data Access: Confidential Data on Untrusted Servers", Int. Conf. on Very Large Data Bases (VLDB), 2002.

[BPS96] Baraani A., Pieprzyk J., Safavi-Naini R. "Security In Databases: A Survey Study", 1996.

[CFZ01] Cherniak M., Franklin M., Zdonick S., "Data Management for Pervasive Computing", Int. Conf. on Very Large Data Bases (VLDB), 2001.

[Ch01] Cherniak M., Franklin M.J., Zdonik S., "Expressing User Profiles for Data Recharging", IEEE Personal Communications, 2001.

[CCB02] Conan D., Chabridon S., Bernard G., "Disconnected Operations in Mobile Environments", Parallel anf Distributed Processing Symposium, 2002.

[DGH87] Demers A., Greene D., Hauser C., Irish W., Larson J., "Epidemic algorithms for replicated database maintenance", ACM Symposium on Principles of Distributed Computing (PODC), 1987.

[EuG02] Eugster P., Guerraoui R., "Probabilistic multicast", IEEE Int. Conf. on Dependable Systems and Networks (DSN), 2002.

[EGH03] Eugster P., Guerraoui R., Handurukande S., Kermarrec A., Kouznetsov P., "Lightweight probabilistic broadcast", ACM Transaction on Computer Systems, 2003.

[FBI02] Computer Security Institute, "CSI/FBI Computer Crime and Security Survey" http://www.gocsi.com/forms/fbi/pdf.html.

[FKV00] Fritsch D., Klinec D., Volz S., "NEXUS – Positioning and Data Management Concepts for Location Aware Applications ", Int. Symp. on Telegeoprocessing, 2000.

[HIL02] Hacigumus H., Iyer B.R., Li C., Mehrotra S., "Executing SQL over Encrypted Data in the Database-Service-Provider Model", ACM SIGMOD, 2002.

[IMO03] A. Imine, P. Molli, G. Oster, and M. Rusinowitch. "Proving Correctness of Transformation Functions in Real-time Groupware", European Conf. on Computer-Supported Cooperative Work, 2003.

[ImN02] Imielinski T., Nath B., "Wireless Graffiti – Data, data everywhere", Int. Conf. on Very Large Data Bases (VLDB), 2002.

[JTK97] Joseph A., Tauber J., Kaashoek M., "Mobile computing with the Rover toolkit", IEEE Transactions on Computers, 46(3), 1997.

[Kum00] Kumar V., "A Timeout-based Mobile Transaction Commitment Protocol", Proc. ADBIS-DASFAA, LNCS vol. 1884, 2000.

[MCE03] Mascolo C., Capra L., Emmerich W., "Middleware for Mobile Computing (A Survey)", Advanced Lectures on Networking - Networking 2002 Tutorials, 2002.

[MaF02] Madden S., Franklin M., "Fjording the Stream: An Architecture for Queries over Streaming Sensor Data", Int. Conf. on Data Engineering (ICDE), 2002.

[MFH02] Madden S., Franklin M., Hellerstein J., Hong W., "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks", Int. Conf on Operating Systems Design and Implementation, 2002.

[MoV00] Momin K.A., Vidyasankar K., "Flexible Integration of Optimistic and Pessimistic Concurrency Control in Mobile Environments", LNCS vol. 1884, 2000.

[NSN97] Noble B., Satyanaratanan M., Narayanan D., Tilton J., Flinn J., Walker, K., "Agile Application-Aware for Mobility", ACM Symp. on Operating Systems Principles, 1997.

[Ora02] Oracle Corp., "*Advanced Security Administrator Guide*", Release 9.2, 2002.

[PiB99] Pitoura E., Bhargava B., "Data Consistency in Intermittently Connected Distributed Systems", Transactions on Knowledge and Data Engineering (TKDE), 11(6), 1999.

[PBV01] Pucheral P., Bouganim L., Valduriez P., Bobineau C., "PicoDBMS: Scaling down Database Techniques for the Smartcard", Very Large Data Bases Journal (VLDBJ), 10(2-3), 2001.

[RBV03] van Renesse R., Birman K., Vogels W., "Astrolabe: A Robust and Scalable Technology for Distributed Systems Monitoring, Management, and Data Mining", ACM Transactions on Computer Systems, 21(3), 2003.

[Sat96] Satyanarayanan M., "Mobile Information Access", IEEE Personal Communications, 1996.

[SCF98] Suleiman M., Cart M., Ferrié J., "Concurrent Operations in a Distributed and Mobile Collaborative Environment", IEEE Int. Conf. on Data Engineering (ICDE), 1998.

[SuE98] Sun C., Ellis C.S., "Operational Transformation in Real-Time Group Editors", ACM Int. Conf. on Computer Supported Cooperative Work (CSCW), 1998.

[SRA01] Serrano-Alvarado P., Roncancio C., Adiba M., "Analyzing Mobile Transactions Support for DBMS", Int. Workshop on Mobility in Databases and Distributed Systems (in DEXA), 2001.

[VCF00] Vidot N., Cart M., Ferrié J., Suleiman M., "Copies convergence in a distributed real-time collaborative environment", ACM Int. Conf. on Computer Supported Cooperative Work (CSCW), 2000.

[WaC99] Walborn G.D., Chrysanthis P.K., "Transaction Processing in PRO-MOTION", ACM Symp on Applied Comupting (SAC), 1999.

[WXC98] Wolfson O., Xu B, Chamberlain S., Jiang L., "Moving Objects Databases: Issues and Solutions", Int. Conf. on Scientific and Statistical Database Management, 1998.