View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by KITopen

Usage of mobile databases for mobile geoscientific applications[1]

Wolfgang Bär and Martin Breunig
Research Center for Geoinformatics and Remote Sensing
University of Osnabrück
Eichendorffweg 30, 49377 Vechta, Germany
Phone: ++49-4441-15289 Fax: ++49-4441-15464
{wbaer, mbreunig}@uni-osnabrueck.de

SUMMARY

Recent improvements in mobile device technology make the vision feasible to have geodata analyzed by users right in the field. However, the current practice in mobile applications only allows to gather new data and to retrieve them for viewing on the mobile device. In the future, update capabilities for retrieved data are indispensable. We regard a mobile geo application as being fully integrated into the enterprise system, if it is capable of working in online and offline mode. In the offline mode, the mobile application has to provide update capabilities on local replicated geodata, which can be automatically incorporated back into the enterprise database. To achieve this we propose the use of mobile databases as the storage backend of mobile applications. In this paper we present our ideas for supporting geoscientific 3D mobile applications in offline mode by the usage of a mobile database system integrated with a service-oriented 3D geodatabase management system. The current state in mobile databases is discussed, which includes research problems and existing systems. A concept for integrating mobile databases is presented that is based on the version management paradigm. Finally, preliminary results demonstrating the evaluation of an experimental prototype system are presented.

KEYWORDS: *3D geodatabase, mobile database, geo-service, version management*

INTRODUCTION

The recent improvements in mobile device technology provided less power consuming processors, integrated wireless network adapters and a radically reduced size and weight of devices. These improvements make the vision of geoscientists conceivable to work a full day outdoors, without carrying a supplementary second battery pack.

On the software side, geo-services for mobile data access are also becoming more and more mature. With the standards of the Open Geospatial Consortium (OGC, 2005), like Web Map Service (WMS) or Web Feature Service (WFS), access to geodata is possible in an interoperable way, also from mobile devices. It seems that the infrastructure is prepared to develop mobile applications being fully integrated into the geospatial enterprise. But besides the access to geodata, for future mobile applications the processing and updating will be a necessity. From a hardware point of view, the processing of retrieved geodata is not anymore a problem for todays mobile devices. But from a software point of view, processing and updating geodata raises several severe problems. We have to distinguish between the support for online and offline updating in mobile applications. The first case is, a good wireless connection assumed, already feasible with the OGC standards, although being mostly restricted to the 2D case of geo applications. As the update capabilities of the OGC services are based on locking the geodata, only short duration update transactions are supported. The second

[1] Research project ”Advancement of Geoservices” funded by BMBF (grant no. 03F0373B et al). The responsibility for the contents of this publication is by the authors.

case of updating in offline mode, however, is not yet solved and will require ongoing research over the next years.

We regard a fully functional mobile application as an application being capable working in both modes: First, in an online mode with geodata available in the enterprise system through services. Second, in an offline mode with the option to incorporate the updated local data, reflecting the field work, back to the enterprise databases.

Therefore modern mobile applications must provide spatial database support and a way for the efficient and consistent storage of geodata on the mobile device (Roddick et al., 2004). For this task the use of mobile database systems, transferring all the known benefits from standard database systems to mobile devices, seems to be the best suited approach. However, just putting a mobile database system onto a mobile device for storing geodata is not a solution to the problem. A mobile database system has to be well integrated with the enterprise database system. Regarding mobile databases as a special type of mobile client to enterprise geodatabase systems, it must, for example, be able to synchronise with the latest data available in the enterprise database system. Another example is the incorporation of updated geodata to the enterprise geodatabases, which means that some kind of conflict detection/resolution and merge capabilities must be provided, as in the meanwhile geodata could also have been updated by other users. To realise this scenario, several problems with mobile database systems have to be investigated, which will be addressed in the following.

Geoscientific applications, e.g. in geology (e.g. Thomsen & Siehl, 2002) and geophysics (e.g. Götze & Lahmeyer, 1988), are intrinsically 3-dimensional. In our earlier work we introduced a "3Dto2D-service" of our service-oriented 3D geodatabase system (Breunig et al., 2004). This service is capable of processing geological 3D models for arbitrary planes into 2D profiles to visualize them on a PDA. In this paper we are presenting our advanced ideas for supporting geoscientific 3D mobile applications in offline mode by the usage of mobile databases and version management. The concepts are implemented as an experimental prototype system. We first show relations to other work. We then discuss the problems arising with the usage of mobile databases and give a short overview about the current situation of existing mobile databases, their functionality and relation to geospatial databases. Afterwards, we describe our concept for integration of mobile databases that is mainly based on the version management paradigm.

RELATED WORK

Problems of mobile databases

The problem of wireless connections is widely known (Forman & Zahorjan, 1994; Satyanarayanan, 1995), however, until today researchers in mobile databases are looking for appropriate solutions (Bernard et al., 2004). Besides the obvious quality problem of wireless connections (bandwidth, access coverage outside urban areas), the problem of high costs and battery consumption has to be taken into account. Therefore one has to deal with the problem of disconnection - not only caused by a technical failure, but also by request of the user, who intends to save money or battery power. Actually it will be the usual case in a mobile application that most of the work will be done in offline mode. We regard this issue not as a problem as we will use mobile databases for supporting exactly this kind of offline work. The quality problem of wireless connections is existent during the time trying to actually retrieve geodata or synchronizing it with the server database system. As this problem is more hardware than software related, we do not propose a special solution, besides providing transactions. Therefore consistent operations between the mobile and server database are guaranteed. For the case of changing wireless cells during a running transaction several mobile transaction models have been proposed (Serrano-Alvarado et al., 2001; Türker & Zini, 2003). However, in the field of geoscientific applications the users in the field do not move quickly. Therefore we make the assumption that there is no change of the wireless cell during the short time of online connection to the remote server database systems.

Editing data on mobile devices from server database systems causes the problem known as long transactions. This problem has already been described in the GIS field by Newell & Batty (1994) and by other authors. The problem of long transactions can simply be explained as the problem of long editing sessions on geospatial databases causing a long time locking of the edited data. Regarding mobile databases as usual clients, every update on the local database has also to be executed on the server side in a transactional way. This imposes a permanent connection to the server and also the same kind of data locking mechanisms as with a usual client update. Today, version management is widely accepted as a solution to the long transaction problem in spatial database systems (Batty, 2002). However, until now, the authors are not aware of any approaches using version management to integrate mobile databases.

Existing mobile databases

Existing mobile databases on the market (Mutschler & Specht, 2003) are usually used together with their server pendants of the same vendor. Therefore the mobile databases are a subset of the functionality of the corresponding enterprise databases. They all provide support for offline work on replicated data from the server side using synchronization capabilities. By now, these products are not extensible by new user defined datatypes, procedures or index structures. As this is a prerequisite for supporting geodata on mobile databases, it is quite obvious that existing commercial mobile databases are not yet suitable for this task. By now they are providing database functionality for relational data models in standard business application areas. Furthermore, the reintegration of mobile data which has been updated during synchronization is mostly based on basic conflict handling mechanisms (Gollmick, 2003). Simple strategies, such as "server side updates win" or "priority – e.g. deletion over update", are not suitable for mobile application scenarios with a high conflict potential. Especially in geology and geophysics the application models are consisting of a low amount of highly complex objects. Therefore, a high conflict potential is the usual case in these geospatial application fields.

On the other side, there are object-oriented database products (like PSEPro® / Fastobjects®) designed to run on mobile devices (PDAs and subnotebooks). They have the ability to provide database functionality for arbitrary complex data models. The problem with this kind of "mobile" databases is that they are stand-alone products not being integrated with their server pendants at all. So there is no standard way getting these database products to act as mobile databases in connection with their server side database systems. Though being feasible, the software developer has to implement the replication and synchronisation engine on his own.

Version management in spatial database systems

For an overview on version management and version models we refer to Katz (1990). Recent work on version management and integration into object-oriented database applications was done by Schönhoff (2002). There are several database vendors providing support for the long transaction problem in geospatial database systems using the version paradigm. The pioneer in this field, however, is Smallworld GIS® with its integrated database which provided version functionality right from the beginning (Newell & Easterfield, 1990). In the meanwhile almost all vendors of geospatial database systems are using the version management approach in some way in their products - e.g. Oracle WorkspaceManager® (Agarwal et al., 2003) or Esri ArcSDE® (Lonski & Parsons, 2002) - to support long running editing sessions. In Batty (2002) it is realized, that there is a strong requirement to be able to carry out update operations in the field and therefore it is important that the version management approach can support the use of detached mobile databases (offline mode). Nevertheless, at the moment no vendor provides a mobile database product for geospatial applications, regardless of a possible integration with server side database systems or the supported data model dimension (2D, 2.5D or 3D).

CONCEPT FOR THE INTEGRATION OF MOBILE DATABASES

In the following we describe our concept for developing a mobile database which is well integrated with our existing 3D geoscientific database system (Breunig et al., 2004).

The primary idea is the integration of a mobile database by version management. In this conception all the updated geodata on the mobile device become a new revision version of the version initially checked out from the server database system. The described prototype system is fully built on object-oriented database technology. For the server database system we use ObjectStore® and for the mobile database PSEPro®. Going this way it is possible to reuse most of the existing code base for the mobile database, while still being able to tweak parts for existing constraints on mobile devices.

Version models are distinguishing between two parts of a versionable object: A generic part being the "design object" and variable parts being the "versions". Hereby the design object gives the semantics of the versionable object and can be seen as the union of all versions. A single version is a concrete representation of the design object to a certain modeling time. The versions of a design object together form a graph-like structure, the version history, where the nodes of the graph are corresponding to the versions and the edges are representing the type of relationship between the versions (revision, alternative and merge). There is only one initial version, called the root version, created when a database object is put under version control. The versions are organized in repositories called workspaces. These workspaces are connected in a hierarchical structure with one root workspace. This root workspace contains the initial version of the design objects of the current database objects put under version control. The versions can be migrated between connected workspaces in the hierarchy. Furthermore, workspaces provide a mechanism to organize the work on versions, for example by working groups, or to restrict access on versions.

The integration of mobile databases described in the following is divided into three parts. First we describe the version management extension of our 3D database system. The second part describes the realization of our mobile database running on a mobile device. Finally, the last part presents the integration of this mobile database into the version management system.

Extending a 3D database system with version management capabilities

Following the development of GeoToolKit (Balovnev et al., 2004) and the 3D database system introduced in (Breunig et al., 2004), we are describing how the 3D database system is extended on-top with version management capabilities. In the first part of the description a generic version management system is presented. The second part deals with the representation of complex 3D objects as versions.

The generic version management extension is motivated by the object-oriented version model of Schönhoff (2002) and provides the management of history graphs of versions and a hierarchy of workspaces as version repositories. The system provides abstract classes and interface definitions (e.g. DesignObject, Version, HistoryObject) to be implemented by the database object types providing version functionality.

Figure 1 gives an overview of the structure of the version management system. The existing 3D database system is called the release database in which the releasable database objects reside. The design objects and their versions exist in the workspace repositories. At the top there is the root workspace (main workspace) with the design objects and their initial versions. These are created, if a database object from the release database is put under version control for further modifications. Inside a workspace there exist only revisions of versions (linear history). Modifications to versions which are meant to provide a rather alternative representation of an object are called alternative versions. Such alternative versions have to be created in a new child workspace with an own revision history. Propagating a later revision of an alternative version back to the parent workspace is called merging. To execute a merge, no conflicts with the latest revision in the parent workspace are allowed. Otherwise a conflict resolution must be done beforehand by the user. This way mature versions can be propagated upwards the workspace hierarchy and finally replace the original database object in the release database.

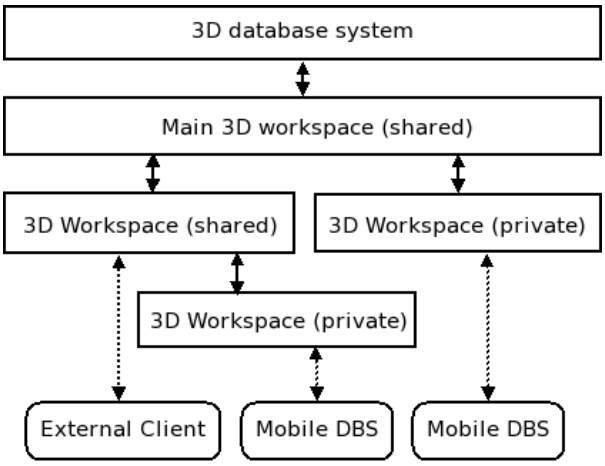

Figure 1: Structure of the version management system.

Besides the concepts of design objects and their versions, the concept of configurations has to be supported in the version management system. Configurations allow grouping specific versions of several design objects. With configurations the notion of 3D models from geology as a consistent set of several design objects can be realized. Therefore they must be extensible to allow forcing constraints on the added versions such as "no geometric intersection between the objects represented by the added versions is allowed". Furthermore, configurations provide a way for batch propagation of a consolidated set of versions between the workspaces.

As the complex 3D objects used in geo applications can internally consist of up to several hundred thousands of simplexes (e.g. triangles, tetrahedrons), the storage of a complete object for each version is unpractical. Therefore, we represent a version as the set of changes to its revision or alternative predecessor in the version history (delta storage). Beside the reduction of used storage, this approach enables efficient algorithms for conflict detection and also for providing change histories. Having used simplicial complexes as the underlying data model of our 3D objects, the changes are represented inside the version management system as additions/deletions of single simplexes, complete components and the associated thematic changes. As the geometric changes are represented by sets of elementary changes (smallest possible unit of change), the algorithms can use rather simple spatial compare processing for the detection of conflicts, new additions or deletions.

Conversion operations between the version representation and the database object representation ensure that all the operations from the geodatabase system can also be applied to the versions of 3D objects. For example, existing services of the geodatabase system can be used, such as the 3Dto2D-Service to create profile sections from different versions of a 3D object.

3D mobile database

As already noticed, the mobile 3D database presented here is built on the code base of the 3D server database system. PSEPro® is used as the underlying mobile object-oriented database system. PSEPro is capable of running on constraint devices (PDAs) and provides a single-user, fully featured database system. Therefore we tested our mobile database also on a PDA.

The mobile database includes the core 3D geometry (0D, 1D, 2D, 3D simplexes) and the object model (simplicial complexes). It is extended to store further version management information for the whole database and each local object. Therefore, each local database object knows its associated design object, version and workspace in the server database. Furthermore, the internal index structure

of the objects is modified for better build up times. To keep the build up time as small as possible, in the mobile database an octree implementation is used as spatial index instead of the R* tree in the server database. This way the build up time gets reduced by a factor of 5, at the cost of longer search times. As the most complex geometric operations (e.g. 3D intersection computations) are heavily dependent on the efficiency of the internal index, they are computed on the server.

Integration of external clients and mobile databases

The integration of both, external clients and mobile databases, into the version management system is realized by providing a set of mobile accessible services. These services enable clients to retrieve meta data for the version management system, retrieving check out and check in versions as usual database objects in XML format and to propagate versions through the workspace hierarchy.

Regarding mobile databases as special clients, they are treated somewhat different from usual LAN based clients. For each mobile database there exists a private workspace inside the version management system. As this workspace is not shared with other clients, there will be no conflicts during the synchronizing step.

A usual working session can be described as follows: First the versions the user is working on during the offline phase are replicated to the mobile database (checkout phase). On the mobile device modifications to the data are realized as local transactions on the mobile database (update/work phase). After finishing work, the updated objects from the mobile database are written back to the version management system as new revisions of the initial checked out versions (checkin phase). Note that the propagation of the new revision versions upwards the workspace hierarchy for incorporation into the main database (propagation phase) is not specific to mobile databases.

EVALUATION

Table 1 shows insert times (in seconds) for objects on the local database running on a Sharp Zaurus® PDA (SL-C760 - XScale 255 CPU, 64 MB RAM). The given times are the average value of 5 test runs.

# Triangles in object	Octree build up time (sec)	Topology build up time (sec)	Commit time (sec)	Total times (sec)
2,500	25	86	19	130
5,000	57	180	42	279
7,500	83	268	60	411
10,000	108	374	76	558
20,000	240	742	166	1,178

Table 1: Build up times for complex 3D objects in the mobile database on a PDA device.

The presented numbers of table 1 are preliminary results from the first evaluation. The only optimization done so far is the octree replacement. Further optimizations will be done in the future, when we will get more feedback from the ongoing evaluation of the mobile database. We judge the total times to be far away from being acceptable. We can conclude that it is indeed possible to run a database for complex 3D geoobjects on such constrained devices, but with the currently available hardware of small mobile devices this can only be achieved with small objects and with unacceptable time delays. There are still several severe limitations like CPU power, slow main memory and slow secondary storage (memory cards). Therefore the main target devices are subnotebooks unless PDAs get more powerful in the future. The difference can be seen by comparing the test results from the PDA with the results on a recent subnotebook (IBM X40, 512 MB RAM) given in table 2 (average of 5 test runs). Beside the clearly better build up times, subnotebooks are also capable of processing rather complex 3D objects – consisting of up to 100 thousand simplexes and more. Further optimization can be achieved by detaching the building up of the topology from the initial object

construction transaction. This could be achieved through starting a low priority background maintenance process after the initial object building, which builds the internal simplex topology. Although this can reduce the object build up times almost by 40 to 50 percent (see Table 1 and 2), operations which require the internal topology, cannot be processed directly after the object build up.

# Triangles in object	Octree build up time (sec)	Topology build up time (sec)	Commit time (sec)	Total times (sec)
10,000	2.2	4.6	1.6	8.4
20,000	4.8	9.2	2.4	16.4
50,000	11.4	23.0	5.2	39.6
100,000	23.1	45.9	11.6	80.6

Table 2: Build up times for complex 3D objects in the mobile database on a subnotebook.

From a database point of view it is regardless whether a PDA or subnotebook is used. The problems, how to deal with disconnection (offline mode) and the ways integrating local changes back to the server database systems, still are the same.

For evaluation purposes also a sample application is under implementation to demonstrate the interaction between mobile and server databases. The application contains a user interface for interacting with the provided version services of the server database system and embeds the mobile database. The application is currently capable of querying the meta data of the remote database, the version management system and version workspaces. Furthermore, a local database can be built from a workspace on the remote database with all existing versions transferred to the local database (see figure 2).

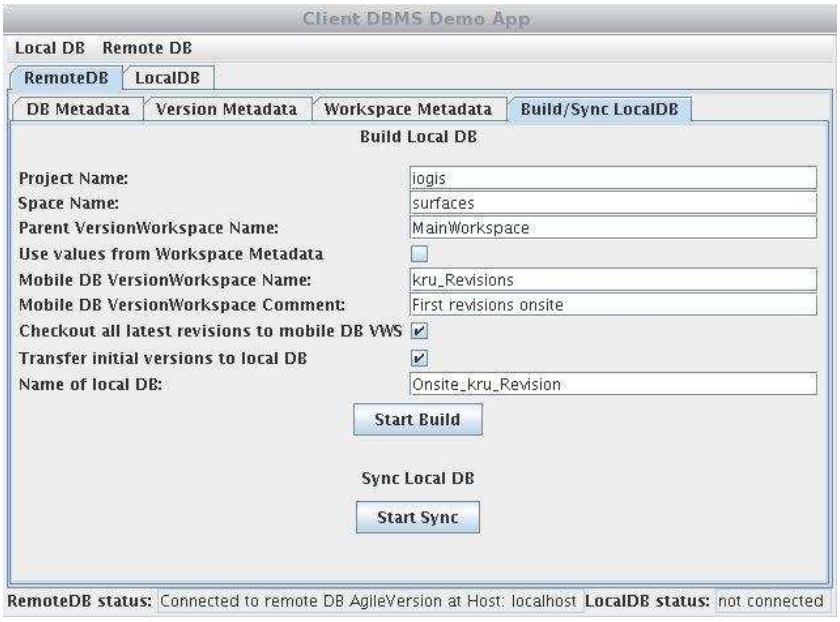

Figure 2*:* Screenshot of the sample application: Panel for building a local database from a remote version workspace and for synchronizing an existing local database.

To work with the local database there is functionality provided for browsing the contents of the local database and displaying the meta data of local objects together with their version information. For the

evaluation of locally updated objects, visualization capabilities are provided to compare objects visually with other revision versions from the remote database (see figure 3). The development of further enhancements like editing of local 3D objects and visualization of conflicts between versions is under construction.

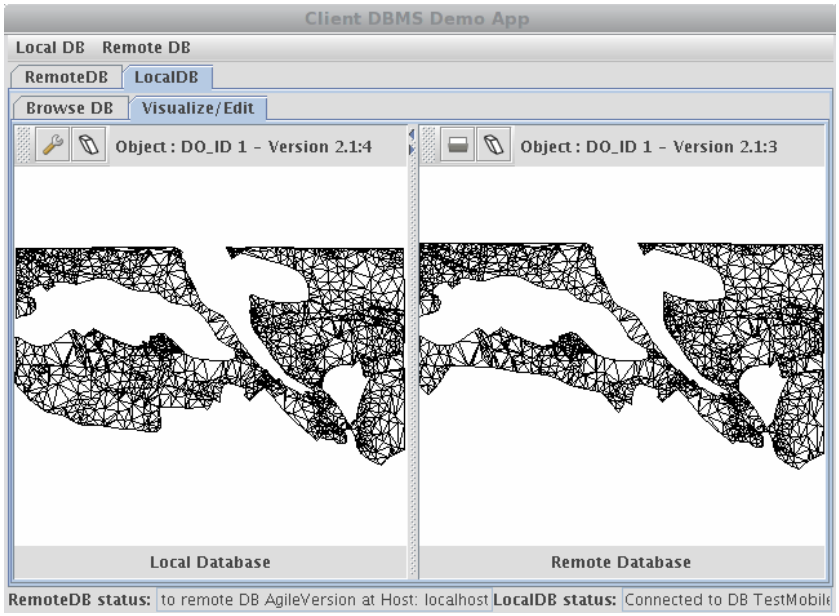

***Figure 3**:* Screenshot of the sample application: Visualisations of a local 3D surface object with predecessor revision version from the remote database server.

CONCLUSION AND OUTLOOK

Todays hardware configurations of small mobile devices, such as PDAs used in mobile geo applications, are not yet suitable for working with large sets of geodata in the field. The usual operation mode at the moment is the online mode where existing data is retrieved or new data is captured. To allow updating geodata also in offline mode, we proposed an approach based on mobile databases and version management. The current state and problems of todays mobile databases have been discussed presenting our idea of how to solve parts of the problem. The approach based on version management enables the user to update replicated geodata in offline mode. The current prototype extends our existing 3D geoscientific database system with version management capabilities. On the client side, a mobile database based on the server database code base has been developed. The mobile database has been integrated into the version management system as a special client.

The main advantage of the version management approach is the ability to always promote the locally made changes back to a server database system, as a new version revision. The decision, about removing, merging or overwriting changes between the version revisions, is made afterwards by the users in cooperation. Therefore, our approach is not well suited for applications, where user interference is not allowed or not wanted. For such scenarios existing automatic resolution schemes – but with the possibility of loosing local updates – are better suited.

The next steps are the ongoing technical evaluation of the version management system and the related mobile database. Current preliminary results show that there is enough room for further research and optimisations. In our future work, we will extend the server side database system with a spatio-temporal data model usable for mobile geological services.

ACKNOWLEDGEMENTS

This is publication no. Geotech-117 of the program Geotechnologien of BMBF and DFG, Grant no. 03F0373B et al.

BIBLIOGRAPHY

Agarwal, S., Arun, G., Chatterjee, R., Speckhard, B. and Vasudevan, R.: Long transactions in an RDBMS. - In 7th Annual Conference and Exhibition: A Geospatial Odyssey, Geospatial Information and Technology Association (GITA), 2003.

Balovnev, O., Bode, T., Breunig, M., Cremers, A.B., Müller, W., Pogodaev, G., Shumilov, S., Siebeck, J., Siehl, A. and Thomsen, A.: The Story of the GeoTookKit - An Object-Oriented Geodatabase Kernel System. Geoinformatica 8:1, Kluwer Academic Publishers, 5-47, 2004.

Batty, P.M.: Version management revisited. – In Proc. of GITA Annual Conference, Florida, 2002.

Bernard, G. et al.: Mobile Databases: a Selection of Open Issues and Research Directions. ACM SIGMOD Record, Vol. 33, No. 2, 6 p., June 2004.

Breunig, M., Bär, W. and Thomsen, A.: Usage of Spatial Data Stores for Geo-Services. – In Proc. 7th AGILE Conference on Geographic Information Science, Heraklion, Greece, 687-696, 2004.

Forman, G. and Zahorjan, J.: The Challenges of Mobile Computing. IEEE Computer, 27 (4), 38-47, 1994.

Gollmick, Ch.: Client-Oriented Replication in Mobile Database Environments. Jenaer Schriften zur Mathematik und Informatik, Math/Inf/08/03, University of Jena, Germany, 2003.

Götze, H.-J. and Lahmeyer, B.: Application of three-dimensional interactive modeling in gravity and magnetics. Geophysics, Vol. 53, No. 8, 1096-1108, 1988.

Katz, R.H.: Towards a Unified Framework for Version Modeling in Engineering Databases. ACM Computing Surveys 22, No. 4, 375-408, 1990.

Lonski, T.E. and Parsons, R.: Transaction Management and Versioning for Enterprise-wide ESRI Implementations. - In Proc. of the ESRI user conference ‘02, 2002.

Mutschler, B. and Specht, G.: Implementation concepts and application development of commercial mobile database systems (in german). – In Proc. Workshop Mobility and Information Systems, ETH-Zürich, Report Nr. 422, 67-76, 2003.

Newell, R.G. and Easterfield, M.: Version Management – the problem of the long transaction. Smallworld Systems Ltd., Technical Paper 4, 1990.

Newell, R.G. and Batty, P.M.: GIS Databases are different. – In Proc. AM/FM International Annual Conf. XVII, Denver, 279-288, 1994.

OGC: Open Geospatial Consortium, http://www.opengeospatial.org, 2005.

Roddick, J.F., Egenhofer, M.J., Hoel, E. and Papadias, D.: Spatial, Temporal and Spatio-Temporal Databases – Hot Issues and Directions for PhD Research. ACM SIGMOD Record, Vol. 33, No. 2, 6 p., June 2004.

Satyanarayanan, M.: Fundamental Challenges in Mobile Computing. ACM Symposium on principles of distributed computing, 7 p., 1995.

Schönhoff, M.: Version Management in Federated Database Systems. DISDBIS 81, Akademische Verlagsgesellschaft (Aka), Berlin, 2002.

Serrano-Alvarado, P. et al.: Mobile Transaction Supports for DBMS: An Overview. Rapport de Recherche, RR 1039-1-LSR 16, Laboratoire LST-IMAG, Grenoble, France, 2001.

Thomsen, A. and Siehl, A.: Towards a balanced 3D kinematic model of a faulted domain – the Bergheim open pit mine, Lower Rhine Basin. Netherlands Journal of Geosciences, 81 (2), 241-250, 2002.

Türker, C. and Zini, G.: A Survey of Academic and Commercial Approaches to Transaction Support in Mobile Computing Environments. Technical report 429, ETH Zürich, Department of Computer Science, 25 p., 2003.