

A System of Adaptable Transactions for The Mobile Computing Environment

Tarcisio da Rocha, Maria Beatriz F. de Toledo

*Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6.176 – 13.083-970 – Campinas – SP – Brazil
{tarcisio.rocha,beatriz}@ic.unicamp.br*

1. Summary

The widespread use of mobile computing imposes a series of obstacles demanding new approaches for the development of distributed applications.

Our work focuses on transaction management with the revision of the traditional transaction model to make it more flexible for the mobile environment. The proposed model integrates the application-aware adaptation paradigm with transaction management. This adaptation is based on the collaboration of applications and the underlying system. The underlying system is responsible for resource monitoring and notification of transactions about the significant changes in the environment. When receiving a notification, the transaction can decide the appropriate reactions to the changes. Two basic actions of adaptation are provided to transactions: the change of the isolation level and/or the operation mode. The resource monitor notifies transactions about variations in resource availability and transactions can decide the proper actions to adapt to changes.

An initiating transaction can specify the necessary resources for its execution. Among the resources which can be monitored we can mention bandwidth, disk space and remaining battery power. If the required resource is out of the specified limits the transaction is notified and can change some of its parameters. The first parameter is isolation level based on the ANSI/ISO SQL-92 pattern. *Level 0* requires just short write locks. *Level 1* requires long write locks. *Level 2* requires short read locks and long write locks and *Level 3* guarantees serializability with long read locks and long write locks. The second parameter is operation mode. There are three modes: *remote*, *local/remote* and *local*. In the *remote* mode, the operations on objects are executed in the machines where objects are located. The access to remote objects will be subject to locking and will obey the isolation level of the transaction. This approach is suitable when the mobile machine is connected and when objects are too large to be

transferred to local cache. In the *local/remote* mode, remote objects are locked but the operations are executed on a local copy. Only at the end of a transaction, the remote objects are updated and the locks released. This approach is suitable when bandwidth is low. In the *local* mode, the operations are executed on local copies and, later, validated with respect to remote copies. This validation consists of checking, for each participant object, if the version of the remote object changed while its local copy was being used. If this is true, the transaction will be aborted and its local results will be discarded. Otherwise, the results of the transaction are transferred to the remote copies. This approach is suitable when the probability of disconnections is high. Both parameters can change during the execution of a transaction.

Finally, a prototype is being developed using Java (J2SDK 1.4.1) and CORBA (Java IDL) with the following components:

- Resource monitor responsible for monitoring resources on behalf of transactions.
- Cache manager responsible for storing objects in the mobile machine.
- Object repository responsible for recovering object states from secondary memory and saving objects into secondary memory.
- Operation queue for queuing operations to be sent to the fixed network when the portable machine is disconnected.