

## TRANSACTION SYNCHRONIZATION IN A DISCONNECTABLE COMPUTER AND NETWORK

This application is a 371 of PCT/US96/11901 filed Jul. 18, 1996. This case claims benefit of provisional application Ser. No. 60/001261 filed Jul. 20, 1995.

### FIELD OF THE INVENTION

The present invention relates to the synchronization of transactions performed on separated disconnectable computers, such as transactions performed on a mobile computer and on a computer network while the mobile computer and the network are disconnected, or transactions performed on separate server computers in a network. More particularly, the present invention relates to the synchronization of transactions when the separate computers are reconnected.

### TECHNICAL BACKGROUND OF THE INVENTION

It is often convenient, and sometimes essential, to carry a computer and selected data while traveling. It may also be convenient or essential to access a computer network using a "mobile computer" such as a laptop, palmtop, notebook, or personal digital assistant. However, different types of mobile computing make very different assumptions about the use and availability of computer networks.

Some mobile computers are not ordinarily connected to a computer network. Like their non-traveling "stand-alone" counterparts, such "walk-alone" computers cannot be connected to a network unless significant hardware or software modifications are made to them or to the network.

"Mobile-link" portable computers are typically connected to a computer network and attempt (with varying degrees of success) to maintain that network connection during mobile use through a wireless link. Typical wireless links use radio waves or infrared light as carriers. Mobile-link computers can be used in a walk-alone mode if the network connection is lost. However, mobile-link systems provide few or no automatic facilities to synchronize the mobile-link computer with the network when the connection is re-established.

"Disconnectable" computers include portable computers that operate in either a walk-alone or a mobile-link mode and provide significant automated facilities for synchronizing operations performed on the mobile computer with operations performed on the network. Disconnectable computers need not be portable. For instance, separate server computers in a wide-area network (WAN) or other network that are connected to one another only sporadically or at intervals may be disconnectable computers.

Unfortunately, conventional disconnectable computers still rely routinely on manually directed file copying to select the data that will be used in the field. Moreover, conventional disconnectable computer systems are not easily extended to support a variety of database formats, and they do not properly handle the situation in which changes to the "same" data are made on both the portable computer and on a network computer during disconnected operation.

For instance, the Coda File System ("Coda") is a client-server system that provides limited support for disconnectable operation. To prepare for disconnection, a user may hoard data in a client cache by providing a prioritized list of files. On disconnection, two copies of each cached file exist: the original stored on the server, and a duplicate stored in the

disconnected client's cache. The user may alter the duplicate file, making it inconsistent with the server copy. Upon reconnection, this inconsistency may be detected by comparing timestamps.

However, the inconsistency is detected only if an attempt is made to access one of the copies of the file. The Coda system also assumes that the version stored in the client's cache is the correct version, so situations in which both the original and the duplicate were altered are not properly handled. Moreover, the Coda synchronization mechanism is specifically tailored, not merely to file systems, but to a particular file system (a descendant of the Andrew File System). Coda provides no solution to the more general problem of synchronizing transactions in a distributed database that can include objects other than file and directory descriptors.

Some approaches to distributed database replication are not directed to mobile computing per se but do attempt to ensure consistency between widely separated replicas that collectively form the database. Examples include, without limitation, the replication subsystem in Lotus Notes and the partition synchronization subsystem in Novell NetWare® 4.1 (LOTUS NOTES is a trademark of International Business Machines, Inc. and NETWARE is a registered trademark of Novell, Inc.).

However, some of these approaches to replication are not transactional. A transaction is a sequence of one or more operations which are applied to a replica on an all-or-nothing basis. Non-transactional approaches may allow partially completed update operations to create inconsistent internal states in network nodes. Non-transactional approaches may also require a synchronization time period that depends directly on the total number of files, directories, or other objects in the replica. This seriously degrades the performance of such approaches when the network connection used for synchronization is relatively slow, as many modem or WAN links are.

Moreover, in some conventional approaches potentially conflicting changes to a given set of data are handled by simply applying the most recent change and discarding the others. Another drawback of several conventional approaches to replication is the requirement they impose that either or both computer systems be locked out of use while the replicas are being synchronized.

Another drawback of conventional disconnected computing approaches is that the location of data on the mobile computer does not always correspond to its location on the network computer. Files may be located in one subdirectory or on one drive during connected operation and in another subdirectory or on another drive during disconnected operation. Thus, the mobile computer does not present the same view of the network when it is disconnected as it does when connected to the network. In addition to creating a risk of confusion and conflicting file versions, these conventional approaches require users to repeatedly reconfigure application programs to look for data in different locations.

Thus, it would be an advancement in the art to provide a system and method for properly synchronizing transactions when a disconnectable computer is reconnected to a network.

It would be an additional advancement to provide such a system and method which identify potentially conflicting database changes and allow their resolution by either automatic or manual means.

It would also be an advancement to provide such a system and method which are not limited to file system operations but can instead be extended to support a variety of database objects.