

METHOD AND SYSTEM FOR SYNCHRONIZING MULTIPLE USER REVISIONS TO A SHARED OBJECT

BACKGROUND OF THE INVENTION

[0001] File sharing applications allow several different users to collectively share information. Many users may access the same file simultaneously. All of the users may view the file but only the first user to access the file has editing privileges. Subsequent users to access the file are blocked from editing the file. Supplying a read-only version of a file to all but one user is inconvenient for large shared files. This is particularly exacerbated if users want to work on a shared file offline. For example, other users may be locked out of the file for a long period of time if the first user to access the shared file is away on a business trip.

SUMMARY OF THE INVENTION

[0002] The present invention is directed to a method and system for synchronizing multiple user revisions to a shared object. The object may be any entity capable of being shared such as a file. Many different users may access, revise and update the same shared object simultaneously through several different transports. The users are not blocked from accessing and revising a shared object when another user has access to the shared object. Any authorized users may simultaneously revise the shared object. Users are not required to be connected to the shared object while making revisions. The revisions may be made offline to a local cached version of the object. The revisions are then synchronized with other user revisions when the shared object is available. Revisions to the shared object are automatically synchronized such that all users can view the revisions to the shared object. Different users may revise the shared object at different times such that multiple versions of the shared object may coexist. The latest version of the shared object is the version that includes the most recent synchronized revisions that are available to other authorized users.

[0003] A conflict may occur when two users revise the same portion of the shared object. The revised portion cannot be synchronized with the shared object if it conflicts with another user's revision to the same portion. The portion of the shared object having the conflicting revision is displayed on a conflict page. The conflict page resembles a corresponding master page of the latest version of the shared file except that the portion of the shared object having the conflicting revision is highlighted and displayed in place of the synchronized revision. A conflict indicator is displayed on the master page of the shared object. The conflict page is displayed alongside the master page when the conflict indicator is selected. The user is presented with both the synchronized state of the master page and the corresponding conflict page. The user may reconcile and merge the conflicting revisions into the master page. Conflicting revisions that are identified as irrelevant may be purged.

[0004] In one aspect of the invention, a revision to a shared object is received. A determination is made whether the revision conflicts with a synchronized revision on a master page of the shared object. The revision is synchronized with the shared object when the revision is determined to be associated with a current version of the shared object and when the revision is determined to be not conflicting with a synchronized revision.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 1 illustrates a computing device that may be used according to an example embodiment of the present invention.

[0006] FIG. 2 illustrates a block diagram of a system for synchronizing multiple user revisions to a shared object, in accordance with the present invention.

[0007] FIG. 3 illustrates a hierarchical graph of linked nodes that indicate different portions of a shared object, in accordance with the present invention.

[0008] FIG. 4 illustrates a master page of a shared object and an associated conflict page, in accordance with the present invention.

[0009] FIG. 5 illustrates a block diagram of a system for synchronizing multiple user revisions to a shared object, in accordance with the present invention.

[0010] FIG. 6 illustrates an operational flow diagram illustrating a process synchronizing multiple user revisions to a shared object, in accordance with the present invention.

[0011] FIG. 7 illustrates an operational flow diagram illustrating a process for reconciling and merging conflicting revisions from multiple users in a shared object, in accordance with the present invention.

[0012] FIG. 8 illustrates an operational flow diagram illustrating a process for synchronizing multiple user revisions to a shared object, in accordance with the present invention.

[0013] FIG. 9 illustrates an operational flow diagram illustrating a process for seamlessly transitioning from asynchronous to synchronous communication modes, in accordance with the present invention.

[0014] FIG. 10 illustrates an operational flow diagram illustrating a process for seamlessly transitioning from synchronous to asynchronous communication modes, in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0015] The present invention is directed to a method and system for synchronizing multiple user revisions to a shared object. The object may be entity capable of being shared such as a file. Many different users may access, revise and update the same shared object simultaneously through several different transports. The users are not blocked from accessing and revising a shared object when another user has access to the shared object. Any authorized users may simultaneously revise the shared object. Users are not required to be connected to the shared object while making revisions. The revisions may be made offline to a local cached version of the object. The revisions are then synchronized with other user revisions when the shared object is available. Revisions to the shared file are automatically synchronized such that all users can view the revisions to the shared object. Different users may revise the shared object at different times such that multiple versions of the shared object may coexist. The latest version of the shared object is the version that includes the most recent synchronized revisions that are available to other authorized users.