

METHOD OF RESOLVING DATA CONFLICTS IN A SHARED DATA ENVIRONMENT

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority under 35 U.S.C. §119 from U.S. Provisional Application Ser. No. 60/060,225 entitled "Structured Workfolder," filed on Sep. 28, 1997, the contents of which is hereby incorporated by reference.

FIELD OF THE INVENTION

This invention relates to a method and system for resolving data conflicts in a shared data environment.

BACKGROUND OF THE INVENTION

In many business transactions, it is often necessary to assemble a complex set of documents in order to complete the transaction. Folders are a common way of organizing such documents. Some of the documents may be available from inception of the transaction, while others may be received or created as the transaction progresses. Certain documents may need to be available prior to commencing work on various transaction-related tasks. In addition, a number of people may need to work on the transaction simultaneously, all of whom need concurrent access to the transaction information.

Documents and data of this type may be stored in a shared, structured data object. These objects generally comprise various sections into which related documents are grouped, one or more forms into which data can be entered, links to other documents, etc. The data object is stored in a manner which allows it to be simultaneously accessed by many different users as needed. Accordingly, it is necessary to prevent conflicts from arising when two or more people simultaneously access and try to update the same data object.

A conventional solution is to "lock" a file or data object when it is opened and thereby prevent other users from modifying it. Although this scheme prevents conflicts between users, it also limits access by other users to a "read-only" mode until the first user is done and the file is unlocked. Some improvement is gained by a granular locking scheme where, for example, individual records in a database or data object are locked as they are opened, while the remaining database records remain unlocked. This scheme can be made arbitrarily more granular by extended the locking to a finer level such as individual fields within a record. A major drawback with locking schemes, at whatever level of granularity, is that they require users to decide a priori, either implicitly or explicitly, which data elements should be reserved for modification. They also are based on the assumption that users have access to a lock manager at the time that the decision is made to update a particular data element. Finally, granular locking schemes are subject to "deadly embrace" in which deadlock situations arise due to overlapping sets of locks. Locking mechanisms do not work well in a distributed processing environment with replicated data objects or in an environment where multiple off-line users need read/write access to the same data objects.

Several conventional products provide some support for distributed information systems with concurrent user access. These products include OPEN/workflow for Windows NT, published by Eastman Software, Microsoft Exchange, published by Microsoft Corporation, and Lotus Notes, pub-

lished by Lotus Corporation. OPEN/workflow provides a distributed information store with replication on demand. A hierarchical lock manager is used to ensure that only one user can open a record for modification at any one time. A master lock manager keeps track of which information store has permission to lock an individual record. The master lock manager is responsible for transferring locking permission between information stores. The information stores each maintain their individual record. The master lock manager is responsible for transferring locking permission between information stores. The information stores each maintain their own individual lock manager to arbitrate access by individual users working with records from that information store. If a user cannot obtain an exclusive lock, then he or she is presented with a 'read-only' copy. Conflicts are prevented since only one user can modify a record at any one time.

Microsoft Exchange also provides a distributed information store with replication. Any number of users may open and modify the same record. However, only the first user to save a modified record can save it in the original record. Subsequent users who try to save their copy of the record are notified that the record has been changed. They are then given the option of either discarding their changes or saving their edited record as a separate file or record in the information store. The software does not attempt to automatically reconcile the changes to the record made by the users, nor does it assist a user in identifying which record elements have conflicting changes. Where two users are working with different information stores and each modifies their own copy of the record, Microsoft Exchange detects the situation during replication. The software then generates a conflict message that contains both versions of the record and sends a notification to the appropriate users. The software makes no attempt to automatically reconcile the contents of the conflicting records nor are the changes analyzed to assist the users in manual reconciliation.

Lotus Notes also supports information sharing in a distributed environment. Information is organized as "forms", where each form is made up of fields that may be of particular data types, such as text, number, time, and rich text. While both distributed and off-line access to shared data is supported via replication, the program does not automatically reconcile conflicting changes made to the same form by different users. Instead, Notes simply keeps both copies of the form and requires a user, such as the system administrator, to later decide which changes are to be kept and which are to be ignored. In a large system, this process creates a significant amount of work and also can reduce the confidence of users of a busy system that a given form is completely up to date.

Accordingly, it is an object of the invention to provide an improved data environment that allows multiple users to simultaneously access and edit the same shared data object without requiring an objects accessed by one user to be locked relative to later users.

It is a further object of the invention to provide a method and system to analyze changes made by several users to a data object and dynamically detect, mediate, and resolve potential data conflicts.

SUMMARY OF THE INVENTION

The above and other objects are accomplished by providing a system in which multiple users are permitted to access and update the same data object, hereafter referred to as a "Workfolder", without the use of a locking mechanism.