DATA STORAGE

The present invention relates to data storage and more particularly to the storage of data files which may be subjected to updating at more than one location.

In any organisation which has data processing capability at different sites, conflict can arise if common data is differently modified at one site compared to another.

A data file at site "A" may be transferred to site "B" and subsequent amendments at each site result in two completely different end files. This may result in different decisions being taken at the two sites based on what is believed to be the same data.

One way to avoid such conflict is for one site, say site "A" to be in absolute control of the data file, to be responsible for all amendments and data input to that file. This complicates procedure. For example, if new data is to be added from site "B", the data must first be keyed at site "B", transferred to site "A" and checked and entered at site "A" before further processing can take place.

Alternatively, site "B" may have direct input capability to data stores at site "A" but this could result in users at both sites updating a data file at the same time resulting in an unrecognisable file output for either party. Further, such an arrangement requires the use of wide area network technology and high speed data links, which may add significant cost penalties. Incompatability of the processing systems at the two sites can also be a problem.

Making data file copies from a master library at "run-time" is one way to ensure that the latest available data file is available. However, if more than one copy is in use, being updated and potentially returned to the master library only amendments at one site may be recorded.

Such conflicts may have serious effects on multi-sited businesses, particularly if data entry on one site (perhaps sales entry) leads into activity on another site (such as purchasing or scheduling).

According to the present invention there is provided a computer system having a plurality of nodes each node comprising one or more processor units and a respective associated data store, at least two of the nodes storing data, which is common to at least one other node, each common data file which is capable of updating by users at more than one node having the data store at one node designated as a master library, the master library including a record of which other, if any, of the plurality of nodes has users permitted to update the file, the system being characterised in that for each data file to which more than one node has update access there is allocated a single data token which is associated with the data file held in the respective data store of the processing unit currently holding updating rights and write access to any data file is only permitted to the processing unit holding the respective associated token.

Preferably, data tokens are passed between nodes of the system and on each transfer of a token between a node the master library node is notified of token transfer and the latest version of the data file is stored in the master library.

The master library may be arranged to cause transfer of the latest version of a data file to all other nodes which require access to that file on receipt of the token at the master library node.

A computer system in accordance with the present invention will now be described by way of example only, with reference to the accompanying drawings of which:

Figure 1 shows a schematic diagram of one embodiment of a typical single node arrangement;

Figure 2 shows schematically file arrangements in the data store of Figure 1;

Figure 3A shows a distributed computer system according to an embodiment of the invention; and

Figures 3B to 3D show data transfers occurring in the distributed data network of Figure 3a.

This Sheet Has Been Replaced



Referring first to Figure 1, there is shown one node of a computer system including a processing unit 1 serving a number of users 11 to 1n. Each of the users 11 through 1n will access the processing unit 1 by way of a personal computer, for example, in known manner.

The processing unit 1 will act as a fileserver in a known manner to transfer data files from a central data store 2 to a requesting user 11 to 1n dependent upon access rights having been granted to the particular user.

As this is one node of the computer system, network connections 31 to 3m are provided to other nodes in the system either through direct modem links or by dial up links using suitably protected data transfer methods. Thus each node in the system has access to all data held in the data stores 2 in all of the other nodes in the system, although access limitations may be applied.

Turning now to Figure 3a, a network having seven nodes A to G is shown. The nodes may be geographically separated with nodes on different continents if required. Each node in the system will have some programs common to all of the nodes and in particular will have an appropriate electronic mail or similar communications program to enable messages to be sent to other nodes and received therefrom.

Referring again to Figures 1 and 3, considering a data file being created by user 11 at node D, the user effects data entry, say, of a customer list with details of the customer's current service and equipment levels. This causes an identifiable file to be created which will be stored in the data store 2 at node D. The file is identified by name in a known manner resulting in the creation of a file header and for the purposes of this description the file is referred to herein as file "A".

Referring also to Figure 2, the file header, apart from identifying file A, includes "master" and "token" indicators and an access permissions listing. On creation of a file (sometimes referred to as a controlled item) to which more than one node has users with write access, both master

and token indicators are set positive. The creating node (in this case node D), that is where a new controlled item is first originated, is automatically designated the master library for the created item.

User 11 having created the item may enter specific access permissions to other users. Thus a named user at node G may be given read and write access to file A and a general read access only given to all other users in the network. The newly created file A is labelled with a time and date stamp, and messages are created by the processing unit 1 at node B and forwarded to all other nodes. In the case of nodes A, B, C, E and F the message is a copy of file A with read only access permission in the header. In the case of node G, the message is a copy of file A with the specified user for write access and read only access for all others in the header. This message transfer is schematically represented in Figure 3b.

Although for smaller systems, it is possible to forward mail messages including the created file, where larger systems are in use, or to save communications time, the created file may be ommitted. Thus, in one method of working, folowing the creation of file A at node D mail messages are sent to the other nodes announcing its creation and availability. The actual file is only transferred on request from one of the other nodes for that file or if the user at node D (who is the default token holder) instructs the transfer of the token to another authorised user.

Now if the authorised user at node G wishes to update file A the processing unit 1 at node G checks the copy of file A held in its data store to ensure that appropriate access permission has been given. If so, the next step is to read the associated token. If the token is not positive in respect of the particular user, then a mail message is sent to the node (node D) which is the master library for file A. This message requests the update token for the specific user.

Assuming that file A is not in use at node D and the token is positive in the node D data store 2, subject to the

current token holder (user) giving permission the node D processing unit 1 sets its associated token to negative, notes the token destination and forwards a mail message to node G. This mail message comprises the latest version of the file together with the data token. Node G sets the appropriate token positive and the authorised user may now update file A in known manner. This transfer is schematically shown in Figure 3c. Node D may send mail messages to each of the other nodes A, B, C, E and F to warn that any copy of file A currently held in their respective data stores 2 may not be the latest version. This message may also identify the current token holder.

Consider now a new file (file B) created at node A, to which users at nodes C, D, F and G have update access. Node A is the master library and the creating user at node A is the default token holder. Initial file transfer occurs in the manner previously described for file A or on request of the system administrator for the other nodes.

Supposing updates of file B require several users to participate, then the following interchanges may occur; Referring to Figure 3d node D forwards a token request (TR1) to node A. This request results in file B and its token being transferred (TV1) in node D in the same manner as described above. Now, if node C requires access then a request (TR2) is sent to node D. Node D forwards the token and its latest version of file B as a mail message (TV2) to node C. Simultaneously, a mail message (CC2) is sent to node A with the latest version of file B attached but without the token.

Node C may pass the token to node G in similar manner (TR3, TV3) with a mail message (CC3) again being sent to node A. Thus, the master library for file B always contains the latest version of that file but users at node A are not entitled to update it. Also it should be noted that the current token holder is known to the master library.

While file B may be available for read-only access at any of the nodes, only node A and the current token holding node, node G, hold the latest version.

On completion of activity at node G, the node G user may decide to return the token to the default token holder at node A. This requires a mail message (TV4) comprising the updated file B (with its time and date stamp) and the token.

When a node, which is acting as master library for a particular file, receives a message including the token it compares records of versions (time and date stamp) held by each of the other nodes in the system. If any node (even those with read-only access) has an obsolete version then the master library node causes a message to be sent to that other node either attaching a copy of the latest file or indicating that the later version of the file is available. In this latter case then a file transfer will only occur if a request is received from the appropriate node or if the token is again transferred from node A.

Thus, for file B, as the latest version is already at nodes G and A, messages are sent to nodes C, D and F. Nodes B and E are assumed not to have users of file B and do not therefore receive an update.

It should be noted that if a user on node A requires access to file B for update purposes whilst the token is held by another user then token requesting to the appropriate node, say node G, will occur in the same manner as previously described and, subsequent to the current user (token holder) granting permission for file transfer the token and latest version are returned from node G to node A. Although imminent updating of file B is likely to occur because node A has received a latest version together with a token being passed to it from another node the procedure outlined above for updating other nodes in the system will be followed.

When more than one authorised user is on the same node, token transfer occurring between users on the same node does not result in messaging to the master library. Messaging to the master library only occurs when the token is

passed between nodes which results in an update message being sent to the master library.

It should be noted that whenever updating of a file is occurring it is on the latest version of the file since only one update copy ever exists. Whenever the token is transferred between nodes, a copy of the file accompanies the token. Of course, whenever the token is transferred within a node, the latest version of the file is already held in the respective data store.

This Sheet Has Been Replaced



CLAIMS

1. A computer system having a plurality of nodes at different locations, each node comprising one or more processor units and data storesd, wherein at least some data stores at different locations contain common data files for which users at different locations have update access, and each common data file has one such data store designated a master library characterised in that for each common data file to which users at more than one location have update access there is allocated a single data token and update access to any data file is only permitted at the node currently holding the respective associated token.

2. A computer system as claimed in claim 1 further characterised in that on each transfer of a data token between users the latest version of the file associated with the respective data token is transferred.

3. A computer system as claimed in claim 2 further characterised in that on transfer of a token between nodes neither of which includes the master library for the respective token, a copy of the latest version of the respective associated file is also transferred to the node which includes the master library.

4. A computer system as claimed in any preceding claim further characterised in that each time a node receives a token from another node in respect of a file for which the receiving node includes the master library, the processor unit forwards an updating indication to all other nodes holding obsolete versions of the received file.

5. A computer system as claimed in any preceding claim further characterised in that each time a token is transferred between users on differing nodes the node including the master library for the respective file



associated with the transferred token forwards a transfer indication to all other nodes holding copies of that file.

6. A computer system as claimed in any preceding claim further characterised in that each file in the system has a designated master library and an associated update token and only the current token holder can authorise the transfer of the token to any other user.

This Sheet Has Been Replaced



