

when the task manager has indicated that a given task 150 is complete. As previously indicated, a succeeding task becomes active when the preceding task is completed. When the task manager of a given task 150 thinks the required output document(s) 140 are complete, the task manager can initiate the task completion process, for example, by issuing a "commit output document" command. As shown in FIG. 8, once the task manager issues a "commit output document" command, detected in step 805, the reviewers defined for the task are retrieved from the task property list 300 during step 810 and asked to perform a document review. When all the reviewers have approved the output document(s) during step 820, the approval will change the status of the output document 140, and the output document 140 will be copied to become the input document 110 for the next task, if appropriate, during step 825. The status of the task is changed to "complete" during step 830, and the appropriate individuals are notified of the task completion during step 835.

[0051] FIG. 9 illustrates the operation of the sound board 900 of FIG. 6 in further detail. As shown in FIG. 9, the sound board 900 consists of a serializer 951 and a broadcaster 953. In accordance with one aspect of the present invention, each user can submit conflicting change requests for an object spontaneously and concurrently. For example, a first user might request that an object is moved to the left while another user might request that the same object is moved to the right. The serializer 951 receives each of the change requests and serializes them, for example, based on an arrival time or a global clock. Serialized requests are then sent to the broadcaster 953 which broadcasts the requests to all users. As discussed above, the change requests can be broadcast to all currently active users in real-time, and can be stored in the addendum database 420 for subsequent access, e.g., by any late arriving users, as would be apparent to a person of ordinary skill in the art.

[0052] The operating system on the terminal of each user can manage the local user interface in a conventional manner and determine when the local user has requested a change to a shared document. When such a change is requested for a shared document, the operating system can relay the change request to the sound board 900. In one exemplary implementation, the initial change requests made by the local user to a shared document are not processed until the broadcast version of the change request is received back from the broadcaster 953. In a further variation, the initial change requests made by the local user to a shared document can be processed immediately and then discarded when the broadcast version of the change request is received back from the broadcaster 953. Other variations are possible, as would be apparent to a person of ordinary skill in the art based on the present disclosure.

[0053] In the example shown in FIG. 9, user 1, user 3, and user 4 send independent change requests to do A, do B, and do C, respectively. These requests are time ordered by the serializer 951 and sent to the broadcaster 953. The exemplary broadcaster 953 broadcasts the change requests based on the order of receipt to all subscribers including the originator of the change request. Thus, each user receives the same sequence of commands.

[0054] FIG. 10 is a flow chart illustrating an exemplary implementation of a conventional token-based document

management system. As shown in FIG. 10, a first user (user 1), such as a member of a project team, desires to make a change to a shared object (step 1010). Thus, a request is made for the corresponding token(s) (step 1020). The token request is transmitted to a centralized document management system that administers the token. If the centralized document management system determines during step 1030 that the token is not available, the user receives an indication during step 1040 that the user must wait for the token to become available. If the centralized document management system determines during step 1030 that the token is available, then the user receives the token during step 1050.

[0055] Thereafter, user 1 is permitted to make any desired changes, and generates one or more command to modify the object associated with the token (step 1060). The command(s) to change the object are sent to the centralized document management system, and is detected during step 1064. The centralized document management system then broadcasts the change command(s) to each of the active users during step 1068. User 1 receives the broadcast change(s) during step 1070 and implements such changes during step 1080. The token-based document management system continues to process such changes that are requested by a user in possession of the token.

[0056] As previously indicated, a user of the token-based document management system will experience a delay (step 1020) before a desired change can be initiated to a shared object. The user must wait until he or she has possession of the token. The token-based document management system is even more complicated in the case of structured tokens. For example, a white board could be a shared object. On the object, a red color pen, a black color pen and an eraser can be used as tools to make changes. In such cases, one shared object can be changed in different ways. Thus, just to prepare one token for the entire white board is insufficient. It is noted that a red color pen and a black pen will not conflict to use at the same time by different users however a pen and an eraser might not be used at the same time. (since there is no consensus as to whether the pen or eraser is stronger if they work on the same spot). This requires the use of a structured token prohibiting pens and erasers to be used simultaneously, while the token should allow the use of different pens simultaneously. This makes the token-based implementation and design even more difficult. Another example of the use of structured tokens is in a spread sheet application, such as Microsoft Excel, where different tokens may be associated, for example, with each cell, row and column of a spreadsheet.

[0057] FIG. 11 is a flow chart illustrating an exemplary implementation of a shared document revision process incorporating features of the present invention. As shown in FIG. 11, a first user (user 1), such as a member of a project team, desires to make a change to a shared object (step 1110). With the present invention, the user can immediately make any desired changes, and generate one or more command to modify the object associated with the token (step 1120).

[0058] The command(s) to change the object are sent to the sound board 900, and is detected during step 1130. The sound board 900 then broadcasts the change command(s) to each of the active users during step 1140. User 1 receives the broadcast change(s) during step 1150 and implements such changes during step 1160.