

window due to proposed changes from the contributing authors in the case of the managing author (i.e., manager mode), and due to proposed changes from the other contributing authors in the case of each contributing author (i.e., contributor mode). For example, when a person opens a document in manager mode, new updates from the contributing authors are determined and displayed in the comparison window, including new tabs, and in the change summary window. When working in the manager mode, updates are only provided if requested by the managing author. So if a new response arrives, it would not appear in the change summary and comparison windows until the managing author requested an update, for example, by selecting a refresh contribution tab in the file menu. The contributor mode handles the updates the same as or similar to the manager mode. The updates from the other contributors must be requested by the contributing author before his/her GUI is updated, for example, by selecting a refresh contribution tab in the file menu. In one embodiment the contributing author must also request an update for the edit window to display the managing author's accepted or rejected changes from the manager's evolving document window.

[0065] An alternative embodiment of the present invention provides a near simultaneous document editing environment. The changes are displayed nearly simultaneously, because as each contributor makes and submits changes, the changes are sent to tabs on the manager GUI. For example, a manager has a word document open in DMS; A contributor goes to DMS and sees a document he/she wants to edit, requests the document, and because document is open to manager, the document is locked and the contributor gets from DMS a replica of the latest version of the locked document. After the contributor edits the replica, the changes are sent directly to the DMS. Since the manager has also a direct connection to the DMS, a message is sent to the manager's GUI to refresh its view, re-query the comparison window tabs, and re-query the change summary tree. A new tab appears on the manager's view with the new contributor's name and a new pane associated with the new tab with the contributor's changes. The tab (an Active X control) will dynamically track future changes, upon submittal by contributor. Hence if an internal contributor sends a new change to the DMS, the manager's tabs will be updated automatically. All other internal contributors will also see the new changes. Thus the concept includes a replica of an original document being changed (and submitted) by a first person and a second (or third, etc.) person gets and makes changes (and submissions) to another replica and the changes then appear on a separate screen, i.e., tab, on the first person's display, i.e., near simultaneous changes are displayed.

[0066] FIG. 9 is simplified block diagram of the document collaboration system of another alternative embodiment of the present invention. FIG. 9 is similar to FIG. 2 except the revisions and responses are maintained in a separate system, i.e., the collaboration server 230 that comprises a database, rather than in the DMS (or local file system). The collaboration server's database is organized in a hierarchical object structure (FIG. 10) and is implemented using a database engine such as that supplied by Objectivity, Inc. of Mountain View, Calif.

[0067] FIG. 10 is an object model of an aspect of the present invention. An applications object 912 is the root object. The applications object 912, created by the manager

application 214, includes a collaborations collection object 914 having a list of one or more collaborations. A collaboration, represented by collaboration object 920, includes a documents collection object 924, where the documents collections object 924 includes one document, where the document is represented by document object 926. In an alternative embodiment the documents collections object 924 includes one or more documents. A document object 926, is associated with an original document initially selected by the managing author from the DMS or local file system or collaboration server. The document object 926, has a revisions collections object 928, where each revisions collections object 928 has zero or more revisions, e.g., revision object 930. A revision object includes either the original document or a modified original document (i.e., an original document with incorporated changes), i.e., an edited version of the original document as it evolves. Each revision object 930 has a responses collection object 934, where each responses collection object 934 includes zero or more responses, e.g., response object 934. Each response object 934 includes a response from a contributing author.

[0068] In an aspect of the present invention an Application Programming Interface (API) is provided to operate on a database having the object model of FIG. 10. The database may be part of a DMS 118, local file system, or collaboration server 230. In one embodiment, Microsoft® Component Object Model (COM) function interfaces are provided for the objects shown in FIG. 10. The COM interfaces are described below for each object of FIG. 10.

[0069] The application object 912 provides the entry point to the API. No other object can be accessed or created without first creating an application object. The COM interfaces associated with the application object are:

[0070] "ConnectToServer(strBootFilePath As String)" that provides a connection to collaboration store and accepts the file path to the boot file of the server.

[0071] "ImportXDF(strCollaborationID As String)" returns the number of collaboration objects in the list.

[0072] "Collaborations As Collaborations" returns the collaborations collection object 914.

[0073] The collaborations collection object 914 provides access to a list of collaboration objects 920 that are currently open, and adds new collaboration objects. Responses from collaborating authors from any collaboration can be merged at this point giving direct access to the relevant response object 934. The COM interfaces are:

[0074] "Add(strCollaborationID As String) As Collaboration" adds an existing collaboration object 920 to the collection 914. This will return the collaboration object 920.

[0075] "Count As Long" returns the number of collaborations in the list.

[0076] "Item(vItem) As Collaboration" returns the collaboration object requested. This can be accessed by either an index or by the Collaboration Document ID.

[0077] "Create(strDocumentID As String, strTitle As String, strCreator As String, varVersionOriginal) As Collaboration" creates a new collaboration based upon an existing document and returns the collaboration object. Note that varVersionOriginal is a BOOLEAN value.