(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 November 2010 (04.11.2010)

PCT

(10) International Publication Number
WO 2010/126804 A2

(51) International Patent Classification:
*G06F 17/21* (2006.01) *G06F 17/00* (2006.01)

(21) International Application Number:
PCT/US2010/032309

(22) International Filing Date:
23 April 2010 (23.04.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/432,817 30 April 2009 (30.04.2009) US

(71) Applicant *(for all designated States except US)*: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: BAILOR, Jonathan B.; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). ROBINS, David B.; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

# (54) Title: FAST MERGE SUPPORT FOR LEGACY DOCUMENTS

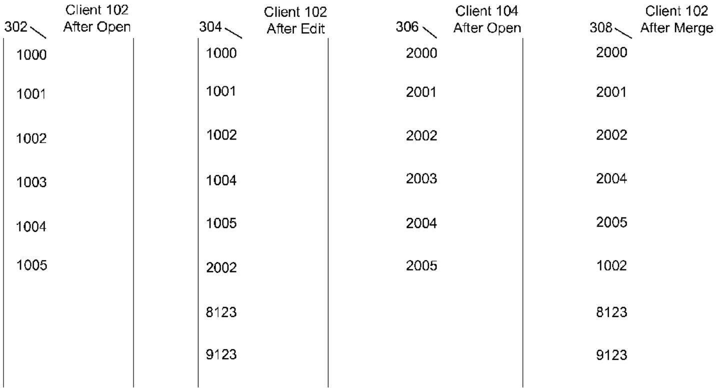

FIG. 3

(57) Abstract: A method is presented for merging document metadata at a client computer. A document obtained from a server computer is opened by a user at the client computer. If the document does not include paragraph identifiers, a paragraph identifier is assigned to each paragraph in the document. When the document is opened by a second user at a second client computer, metadata for the document is received from the server computer. A determination is made whether to merge the metadata received from the server computer with current metadata for the document. When it is determined that the current metadata and the metadata received from the server computer should be merged, paragraph identifiers are adjusted in the document based on the metadata received from the server computer.

# FAST MERGE SUPPORT FOR LEGACY DOCUMENTS

## BACKGROUND

**[0001]** Network-based document collaboration systems permit multiple users to simultaneously access and coauthor a document. Although multiple users are permitted to coauthor a document, in order to prevent conflicts, the system may restrict the areas in a document in which each user can make edits to minimize chances for conflicting changes when the copies of the documents are merged.

**[0002]** Legacy systems typically may not allow for the coauthoring of documents created on the legacy systems. Documents created on these systems may therefore not be configured properly to support coauthoring on document collaboration systems. This can result in undesired conflicting edits.

## SUMMARY

**[0003]** Embodiments of the disclosure are directed to a method for merging document metadata at a client computer. A document is obtained from a server computer by a user at the client computer, and the document is opened at the client computer. In response to opening the document at the client computer, a determination is made as to whether the document includes unit identifiers. When it is determined that the document does not include unit identifiers, a unit identifier is assigned to each unit in the document.

**[0004]** A first message is received from the server computer providing notification that at least one other user is authoring the document. In response to receiving the first message from the server computer providing notification that at least one other user is authoring the document, a second message is sent to the server computer, the second message including first metadata for the document. In response to sending the second message to the server computer, a third message is received from the server computer, the third message including second metadata for the document. In response to receiving the third message from the server computer, a determination is made whether the first metadata and the second metadata should be merged. When it is determined that the first metadata and the second metadata should be merged, unit identifiers are adjusted in the document based on the second metadata.

**[0005]** The details of one or more techniques are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of these techniques will be apparent from the description, drawings, and claims.

## DESCRIPTION OF THE DRAWINGS

**[0006]** Figure 1 shows an example system for fast merge support of legacy documents.

**[0007]** Figure 2 shows example modules of an example client computer of Figure 1 that implements a capability for fast merge support of legacy documents.

**[0008]** Figure 3 an example of document identifier renumbering during a metadata merge operation.

**[0009]** Figure 4 shows a flowchart of a method for merging document metadata at a client computer.

**[0010]** Figure 5 shows a flowchart of a method for determining whether metadata should be merged at a client computer.

**[0011]** Figure 6 shows example components of the client computer of Figure 2.

## DETAILED DESCRIPTION

**[0012]** The present application is directed to systems and methods for creating unit identifiers for legacy documents created on legacy network-based shared document systems that do not provide unit identifiers.

**[0013]** In example embodiments, a document is broken into units. Units can be any subpart of a document, for example sentences, paragraphs, headings, drawings, tables, etc. A document may contain only one unit, for example a single table, sentence, or multiple units, such as multiple tables, sentences, or paragraphs. For the example embodiments disclosed herein, a document unit is a paragraph of the document.

**[0014]** Legacy documents are those documents that were created by legacy systems that do not provide for coauthoring. The legacy documents therefore do not include paragraph identifiers. Paragraph identifiers are created for a legacy document when the document is saved on a document collaboration system or when two or more authors edit the document on the document collaboration system.

**[0015]** Because paragraph identifiers are created for a legacy document in this manner, when two or more authors using the document collaboration system access the same legacy document, paragraphs currently being edited by one of the authors are identified and paragraph locks are propagated to the other authors. The paragraph locks prevent the authors that are not currently editing a locked paragraph from editing the locked paragraph until the lock is removed. In this disclosure, an author is defined as a user who is creating or editing a document.

**[0016]** The systems and methods also provide for comparing and merging paragraph identifiers in documents and for providing paragraph identifier merging and paragraph lock propagation. A unique set of paragraph identifiers is created each time an author accesses a legacy document that does not include paragraph identifiers. When a second author accesses the same legacy document, the paragraph identifiers are compared and merged so that each author has a synchronized set of paragraph identifiers for the legacy document.

**[0017]** Figure 1 shows an example system 100 that supports fast merging of legacy documents. The system 100 includes clients 102, 104, a server 106 and a network 108. More or fewer clients and servers can be used. In this disclosure the terms client and client computer are used interchangeably and the terms server and server computer are used interchangeably.

**[0018]** Among the plurality of information stored on the clients 102, 104 is a client operating system ("OS") and client applications. The client OS is a program that manages the hardware and software resources of the client system. The client applications utilize the resources of the clients 102, 104 to directly perform tasks specified by a user, for example a user who is authoring a document. For example, the clients 102, 104 include one or more software applications, such as word processing programs, that are used to create and edit document files. One example of such an application is Microsoft Word from Microsoft Corporation of Redmond, Washington. Other examples of such applications are also applicable.

**[0019]** Server 106 is a file server that is accessible through a network. Server 106 stores a plurality of files. These files can include both software applications and documents, as described further herein. Server 106 controls access to the documents stored by the server 106.

**[0020]** In example embodiments, the server 106 can be located within an organization or can be part of a document collaboration system. An example document collaboration system is a SHAREPOINT® team services portal server services provided by Microsoft Corporation. An example shared document server is Microsoft Office SharePoint Server 2007 provided by Microsoft Corporation. Other configurations can be used.

**[0021]** In example embodiments, paragraph identifiers are merged and paragraph locks are propagated through a metadata channel 111 that is separate from a data channel 113 used to save the edited contents of the document. By transmitting metadata separately from document content, authors are able to receive paragraph locks fast, independent of a content save operation. Because the paragraph locks prevent two or more authors from editing the same content simultaneously, the integrity of the coauthoring process is enhanced.

**[0022]** Figure 2 shows a detailed view of the logical modules of client computer 102. Client computer 102 includes example document processing module 202, example metadata processing module 204 and example metadata merge module 206. The example document processing module 202 supports the creation, editing and saving of document content, such as text, headers, tables, illustrations, etc. The example document processing module 202 also supports the processing of information about document content, known as metadata. The example document processing module 202 implements a word processing application, for example the Microsoft Word word processing application from Microsoft Corporation.

**[0023]** The example metadata processing module 204 processes the metadata in the document. Some examples of metadata are the document name, date of creation or last edit, document identifier, revision identifier, paragraph identifiers, paragraph range, and paragraph locks. Other kinds of document metadata are possible.

**[0024]** A paragraph identifier is a number that identifies each paragraph in a document. In example embodiments, paragraph identifiers may include identifiers for sub-units of a document such as headers and footers in addition to actual paragraphs.

**[0025]** A paragraph lock is metadata that indicates that an author is editing a paragraph of a document. In order to avoid conflicts, only one author is permitted to edit the same content at the same time. When one author edits a paragraph of a document, a paragraph lock is received for that paragraph for all other authors that have the document open. The document lock prevents these other authors from editing the paragraph until the lock is removed.

**[0026]** Additional details regarding such locks can be found in U.S. Patent Application Serial No. 12/145,536 filed on June 25, 2008, the entirety of which is hereby incorporated by reference.

**[0027]** When an author on client computer 102 opens a legacy document that does not include paragraph identifiers, the example metadata processing module 204 generates a document identifier for the document. The document identifier is typically a number that is randomly generated from a specific number range. For example, the document identifier may be a 32-bit number that is in the range from 1 to 0x7ffffff.

**[0028]** The example metadata processing module 204 assigns the document identifier to the first paragraph in the document. The remaining paragraphs are then sequentially numbered from the document identifier. For example, if there are five paragraphs in the document and the randomly generated document identifier is 1000, the five paragraphs are assigned paragraph identifiers 1000, 1001, 1002, 1003 and 1004, respectively.

**[0029]** The example metadata processing module 204 also identifies as metadata the next paragraph number in the paragraph identifier sequence. In this example, the next paragraph identifier is 1005. Although the paragraph identifiers are sequentially generated when a document is initially opened, paragraph identifiers for any paragraphs that may be added as a result of editing the document are randomly generated.

**[0030]** When the author on client 102 opens the document, the example metadata processing module 204 also obtains a revision identifier for the document. Each document stored in the example document collaboration system includes a revision identifier that identifies a version number for the document. In some example embodiments, the revision identifier may be part of a file system identifier for the document. The example metadata processing module 204 keeps track of the revision identifier as metadata.

**[0031]** Because at this time, the author on client 102 is the only author that has opened the document, the document does not contain any paragraph locks. The author on client 102 is free to edit the document at will, editing existing paragraphs, deleting paragraphs, adding new paragraphs, making formatting changes, etc. Until the author on client 102 saves these changes, the server 106 does not know about the changes.

**[0032]** When an author on client 104 opens a copy of the same document, server 106 presents the author on client 104 with a version of the document that does not include the changes the author on client 102 made to the document if the author on client 102 has not saved the document before client 104 opens the document.

**[0033]** When the document is opened on client 104, the example metadata processing module 204 on client 104 randomly generates a document identifier for the version of the document opened at client 104. The example metadata processing module 204 on example client 104 assigns the document identifier to the first paragraph of the document opened on client 104 and sequentially assigns paragraph numbers to the remaining paragraphs of the document opened on client 104.

**[0034]** Because the document identifiers are randomly generated, the document and paragraph identifiers for the copies of the document opened at clients 102 and 104 are different. In order to propagate paragraph locks to the two opened documents, the paragraph identifiers for each opened document must be merged and synchronized. This is because paragraph locks are associated with specific paragraphs. Therefore, paragraphs having common content on multiple copies of opened documents are required to have to have identical paragraph identifiers.

**[0035]** The example metadata merge module 206 compares and synchronizes paragraph identifiers between documents. When the author on client 104 opens the document, server 106 informs the author on client 104 that there is at least one other user already authoring the document. Server 106 also informs the author on client 102 that another author has opened the document. When the author at client 102 discovers that another author has opened the document, the example metadata merge module 206 on client 102 sends metadata to server 106. Similarly, when the author at client 104 discovers that another author has opened the document, the example metadata merge module 206 at client 104 sends metadata to server 106 using the metadata channel 111.

**[0036]** The example metadata sent to server 106 by clients 102 and 104 includes three metadata components – the document identifier, the next paragraph identifier and the revision identifier. In example embodiments, these three components are called a metadata seed. When server 106 receives the metadata seed from clients 102 and 104, server 106 determines which metadata seed arrived first. The metadata seed that arrived at server 106 first is considered the winner. The winning metadata seed information is then propagated to the other authors. For example, if server 106 receives the metadata seed from client 104 first, the metadata seed information on client 102 is merged with the metadata seed information from client 104.

**[0037]** Figure 3 shows an example of how the example metadata merge module 206 merges metadata using metadata from two authors. When an author on client 102 initially opens a document that doesn't include paragraph identifiers (e.g., a document created on a legacy system that does not support coauthoring), metadata operations merge module 204 assigns paragraph identifiers to each paragraph of the document. An example set of paragraph identifiers is shown for example document copy 302. In this example, there are six paragraphs in the document with paragraph identifiers 1000, 1001, 1002, 1003, 1004 and 1005, respectively.

**[0038]** During the editing of the document, the author on client 102 deletes paragraph 1003 and adds paragraphs with paragraph identifiers 2002, 8123 and 9123. This is shown in example document copy 304. The author on client 102 is able to modify, add, and/or delete any paragraph in the document because currently there are no other users authoring the document so there are no paragraph locks on the document. The added paragraphs have paragraph identifiers that are randomly generated, as described above.

**[0039]** After the author on client 102 makes the changes to the document shown in example document copy 304, the author on client 104 opens a copy of the document. A copy of the document is obtained from server 106. Because the changes to the document made by the author on client 102 have not yet been saved to server 106, the document opened by the author on client 104 contains the same content as shown in example document copy 302.

**[0040]** When the document is opened on client 104, the metadata operations module 204 on client 104 randomly generates a document identifier for the document opened at client 104. In this example, the document identifier is 2000. The example metadata operations module 204 on client 104 assigns the document identifier 2000 to the first paragraph of the document opened on client 104. The example metadata operation module 204 on client 104 then sequentially numbers the remaining paragraphs in the document. The resulting paragraph numbering is shown in example document copy 306.

**[0041]** At some given interval, clients 102, 104 both synchronize the metadata associated with the document. This synchronization happens over the metadata channel 111, as described above. In example embodiments, the synchronization interval can be set to happen at a specific period, such as every 2 second, 5 second, 10 seconds, 20, seconds, 30 seconds, 1 minute, 5 minutes, 10 minutes, or 30 minutes. Other intervals can be used.

**[0042]** In the example, the metadata seed from client 104 is received at server 106 first. When server 106 determines that the metadata seed from client 104 was received first and is the winning seed, server 106 sends the metadata seed received from client 104 to client 102. When client 102 receives the metadata seed from server 106, the example metadata merge module 206 on client 102 merges the metadata seed received from server 106 with the paragraph metadata on client 102. The result of the merge is shown in example document copy 308.

**[0043]** Example document copy 308 shows that as a result of the merge, paragraphs identifiers 1000, 1001, 1002, 1004 and 1005 (from example document copy 304) are changed to 2000, 2001, 2002, 2004 and 2005, respectively. Because the document identifier of the winning seed is 2000, the example metadata merge module 206 on client 102 renumbers the paragraphs in the paragraph range determined from the metadata seed received from server 106. The renumbering makes the paragraphs in the paragraph range consistent with the document identifier 2000. The paragraph range can be determined from the metadata seed because the metadata seed includes the document identifier (which is also the first paragraph of the paragraph range) and the next paragraph identifier (which is the next sequential paragraph identifier following the last paragraph identifier in the paragraph range) and because all the paragraphs in the paragraph range are sequentially numbered. Therefore, the first paragraph in the document opened at client 102 is renumbered from 1000 to 2000. The paragraph range includes paragraphs 1000 through 1005. Therefore, paragraph 1001 is renumbered to 2001, paragraph 1002 is renumbered to 2002, paragraph 1004 is renumbered to 2004 and paragraph 1005 is renumbered to 2005. As discussed earlier, paragraph 1003 was deleted.

**[0044]** The metadata merge module 206 on client 102 also determines all paragraph identifiers the document opened at client 102 that are not included in the paragraph range. This includes paragraphs with paragraph identifiers 2002, 8123 and 9123 that were added to example document copy 304. Because added paragraph 2002 conflicts with renumbered paragraph 2002 (original paragraph 1002), the example metadata merge module 206 renumbers added paragraph 2002 to 1002. The example metadata merge module 206 renumbers added paragraph 2002 to 1002 because paragraph identifier 1002 is no longer used in the document and, therefore, there is no conflict if paragraph 1002 is used. The example metadata merge module 206 does not change paragraph identifiers 8123 and 9123 because there are no conflicts caused by these paragraph identifiers.

**[0045]** When the metadata merge is completed, paragraphs with common content for the document opened on client 102 and for the document opened on client 104 have identical paragraph identifiers. For example, if before the merge paragraphs having identifiers 1000, 1001, 1002, 1004 and 1005 for the document opened on client 102 have common content with paragraphs having identifiers 2000, 2001, 2002, 2004 and 2005, respectively for the document opened on client 104, after the merge, these paragraphs on both client 102 and client 104 have paragraph identifiers 2000, 2001, 2002, 2004 and 2005. This permits paragraph locks to be assigned to these paragraphs. For example, when an author at client 102 edits paragraph 2004, a paragraph lock is transmitted to the document at client 104, preventing client 104 from editing paragraph 2004. The paragraph lock assures that multiple authors cannot edit the same paragraph simultaneously.

**[0046]** When the author on client 102 performs a save operation, the content changes for the document opened on client 102 are sent to server 106 through the data channel 113. In addition to transmitting content changes, the save operation also removes any paragraph locks associated with the paragraphs previously edited on client 102. The change in locks is communicated through the metadata channel 111.

[0047] The saved content and the removal of the paragraph locks are propagated to all authors that are currently authoring the document. This permits the authors to edit paragraphs that were previously locked. However, once an author starts editing a paragraph, a new paragraph lock is created and propagated to all other authors that are currently editing the document through the metadata channel 111.

[0048] Once the paragraph identifiers are synchronized, further changes to the document by either client 102 or client 104 can be synchronized using the paragraph identifies in the manner described in U.S. Patent Application Serial No. 11/951,973 filed on December 6, 2007, the entirety of which is hereby incorporated by reference.

[0049] The above discussion of metadata merging describes two authors. In other example embodiments, more than two authors may edit the same document. The same metadata merge operations apply for example embodiments having more than two authors.

[0050] Figure 4 shows an example flowchart of a method 400 for merging document metadata at a client computer. At operation 402, an author on example client 102 obtains a document from example server 106 and opens the document on client 102. Server 106 is a server such as Microsoft SharePoint Server 2007, used with an Internet-based document collaboration system such as Microsoft SharePoint. Client 102 opens the document with a word processing program such as Microsoft Word. Example method 400 assumes that at the time the author at client 102 opens the document, no other authors have the document open.

[0051] At operation 404, client 102 determines whether the document includes paragraph identifiers. Typically, legacy documents stored on server 106 do not include paragraph identifiers because the legacy documents are created by word processing applications that do not support coauthoring or include paragraph identifiers. If the document does include paragraph identifiers, client 102 can start editing the document at operation 406.

[0052] If the document does not include paragraph identifiers, paragraph identifiers are assigned to the document at operation 408. The example metadata operations module 204 randomly generates a document identifier for the document. The document identifier, typically a 32-bit number with a range from 1-0x7fffffff, is then assigned to the first paragraph of the document. Then, the remaining paragraphs are sequentially numbered from the document identifier.

[0053] At operation 410, the author on client 102 proceeds to edit the document. Because the author on client 102 is the only author that has opened the document, the author on client 102 can edit any paragraph of the document and can also add and delete paragraphs.

**[0054]** At operation 412, client 102 receives a message from server 106 that provides notification to client 102 that at least one other author, for example an author on client 104, has opened the document. When client 102 receives notification that at least one other author has opened the document, at operation 414 the example metadata operations module 204 on client 102 sends first metadata to server 106. The first metadata includes a metadata seed for the document open on client 102, including the document identifier, a next paragraph identifier and a revision identifier.

**[0055]** At operation 416, client 102 receives a message from server 106 that includes second metadata. The second metadata originates from one of the other authors who is authoring the document, for example the author on client 104. The second metadata represents the winning metadata seed received at server 106. The winning metadata seed is the first metadata seed received at server 106 from the at least one other authors that are authoring the document. The second metadata includes the document identifier, next paragraph identifier and revision identifier from the client that sends the winning metadata seed to server 106, for example client 104.

**[0056]** At operation 418, client 102 determines whether the first metadata and the second metadata should be merged. The steps involved in determining whether the first metadata and the second metadata should be merged are shown in Figure 5 and discussed later in this disclosure. If it is determined that the first metadata and the second metadata should not be merged, at operation 420 the author at client 102 continues to edit the document.

**[0057]** If it is determined that the first metadata and the second metadata should be merged, at operation 422 the example metadata merge module 206 on client 102 merges the first metadata and the second metadata. The metadata is merged by adjusting paragraph identifiers in the document opened by the author on client 102. The operation of adjusting paragraph identifiers includes changing the paragraph identifiers for all paragraphs in the document opened on client 102 that have common content with corresponding paragraphs in the document opened on client 104. The paragraph identifiers on the document opened on client 102 are changed to corresponding paragraph identifiers indicated by the document identifier and the next paragraph identifier included in the second metadata. For example, if paragraphs 1-3 on the document opened on client 102 have common content with paragraphs 1-3 on the document opened on client 104 and if the document identifier for the document on client 102 is 1000 and the document identifier in the second metadata is 2000, the first three paragraphs in the document on client 104 are renumbered to 2000, 2001 and 2002.

**[0058]** In addition, if it is determined that paragraphs added to the document opened on client 102, but not saved, have paragraph identifiers in the paragraph range indicated by the second metadata, the example metadata merge module 206 changes these paragraph identifiers so that they are unique in the document.

**[0059]** Figure 5 shows an example flowchart of a method 418 for determining whether metadata should be merged at a client computer. The method is implemented when an author at a client computer, for example client computer 102, edits a document that includes a first metadata seed and the client computer receives a second metadata seed from a server computer. The second metadata seed represents metadata from a second author that has opened the document. The second metadata seed represents the first metadata seed received at the server computer from one or more authors that have opened and are authoring the document. The metadata seeds include a document identifier, a next page identifier and a revision identifier.

**[0060]** At operation 502, a determination is made whether the revision identifier from the first metadata seed matches the revision identifier from the second metadata seed. The revision identifier indicates the version of the document. In order to merge metadata, the documents must be at the same revision level. This is because when each author initially opens the document, the content of both documents must be the same. When it is determined that the revision identifier from the first metadata seed does not match the revision identifier from the second metadata seed, at operation 504 metadata is not merged and the author at client 102 continues editing.

**[0061]** When it is determined that the revision identifier from the first metadata seed matches the revision identifier from the second metadata seed, at operation 506 a determination is made whether the document identifier from the first metadata seed matches the document identifier from the second metadata seed. When the document identifiers match, it indicates that documents have already been synchronized and that a metadata merge is not necessary. In this case, at operation 508 metadata is not merged and the author at client 102 continues editing.

**[0062]** When it is determined that the document identifier from the first metadata seed does not match the revision identifier from the second metadata seed, at operation 510, a determination is made whether the size of the paragraph range as determined from the first metadata seed matches the size of the paragraph range as determined from the second metadata seed. When it is determined that the paragraph ranges are not the same, at operation 512 metadata is not merged and the author at client 102 continues editing.

**[0063]** When it is determined that the paragraph range as determined that the paragraph ranges are the same, at operation 514 the first metadata seed and the second metadata seed are merged. The merging of the first metadata seed and the second metadata seed comprises adjusting paragraph identifiers in the document opened on client 102 per operation 422, as discussed.

**[0064]** With reference to Figure 6, example components of client 102 are shown. In example embodiments, the client 102is a computing device, such as a desktop computer, laptop computer, personal data assistant, or cellular device. The client 102 can include input/output devices, a central processing unit ("CPU"), a data storage device, and a network device.

**[0065]** In a basic configuration, the computing device 102 typically includes at least one processing unit 602 and system memory 604. Depending on the exact configuration and type of computing device, the system memory 604 may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. System memory 704 typically includes an operating system 606 suitable for controlling the operation of a networked personal computer, such as the WINDOWS® operating systems from MICROSOFT CORPORATION of Redmond, Washington or a server, such as Windows SharePoint Server 2007, also from MICROSOFT CORPORATION of Redmond, Washington. The system memory 604 may also include one or more software applications 608 and may include program data.

**[0066]** The computing device 102 may have additional features or functionality. For example, the computing device 102 may also include additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in Figure 6 by removable storage 610 and non-removable storage 612. Computer storage media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. System memory 604, removable storage 610 and non-removable storage 612 are all examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computing device 102. Any such computer storage media may be part of device 102. Computing device 102 may also have input device(s) 614 such as keyboard, mouse, pen, voice input device, touch input device, etc. Output device(s) 616 such as a display, speakers, printer, etc., may also be included. These devices are well known in the art and need not be discussed at length here.

**[0067]** The computing device 102 may also contain communication connections 618 that allow the device to communicate with other computing devices 620, such as over a network in a distributed computing environment, for example, an intranet or the Internet. Communication connection 618 is one example of communication media. Communication media may typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. The term computer readable media as used herein includes both storage media and communication media.

**[0068]** The various embodiments described above are provided by way of illustration only and should not be construed to limiting. Various modifications and changes that may be made to the embodiments described above without departing from the true spirit and scope of the disclosure.

# CLAIMS

1. A method (400) for merging document metadata at a client computer (102), the method comprising:

obtaining a document from a server computer (106) at the client computer (102), the document being requested by a user at the client computer (102);

opening the document at the client computer (102);

in response to opening the document at the client computer (102), determining whether the document includes unit identifiers;

when it is determined that the document does not include unit identifiers, assigning a unit identifier to each unit in the document;

receiving a first message from the server computer (106) providing notification that at least one other user is authoring the document;

in response to receiving the first message from the server computer (106) that at least one other user is authoring the document, sending a second message to the server computer (106), the second message including first metadata for the document;

in response to sending the second message to the server computer (106), receiving a third message from the server computer (106), the third message including second metadata for the document;

in response to receiving the third message from the server computer (106), determining whether the first metadata and the second metadata should be merged; and

when it is determined that the first metadata and the second metadata should be merged, adjusting unit identifiers in the document based on the second metadata.

2. The method of claim 1, wherein the first metadata includes a first document identifier, a first next unit identifier and a first revision identifier and wherein the second metadata includes a second document identifier, a second next unit identifier and a second revision identifier.

3. The method of claim 2, wherein a first unit identifier range is determined from the first document identifier and wherein a second unit identifier range is determined from the second metadata, the first unit identifier range including a first identifier and a last identifier and the second unit identifier range including a first identifier and a last identifier, the first identifier in the first unit identifier range having an initial value equal to the first document identifier and the last identifier in the first unit identifier range having a value equal to one less than the first next unit identifier, the first identifier in the second unit identifier range have an initial value equal to the second document identifier and the

last identifier in the second unit identifier range have a value equal to one less than the second next unit identifier.

4. The method of claim 3, wherein determining whether the first metadata and the second metadata should be merged comprises determining whether the first revision identifier matches the second revision identifier.

5. The method of claim 4, further comprising, when it is determined that the second revision identifier matches the first revision identifier, determining whether the first document identifier matches the second document identifier.

6. The method of claim 5, further comprising, when the first document identifier does not match the second document identifier, determining whether the size of the first unit identifier range matches the size of the second unit identifier range.

7. The method of claim 6, further comprising, when the size of the first unit identifier range matches the size of the second unit identifier range, determining that the first metadata and the second metadata should be merged.

8. The method of claim 3, wherein the first document identifier is randomly generated, wherein each unit in the first unit identifier range is sequentially numbered starting from the first document identifier, and wherein each unit in the second unit identifier range is sequentially numbered starting from the second document identifier.

9. The method of claim 8, wherein adjusting unit identifiers in the document based on the second metadata comprises changing the first identifier in the first unit identifier range to the second document identifier.

10. The method of claim 9, further comprising changing unit identifiers for units included in the first unit identifier range to corresponding unit identifiers included in the second unit identifier range.

11. The method of claim 10, further comprising determining whether one or more unit identifiers from units not included in the first unit identifier range are included in the second unit identifier range.

12. The method of claim 10, further comprising, when it is determined that one or more unit identifiers from units not included in the first unit identifier range are included in the second unit identifier range, changing the one or more unit identifiers from units not included in the first unit identifier range to a unit identifier included in the first unit identifier range.

13. A computing device (102), comprising:

a processing unit (602);

a system memory (604) connected to the processing unit (602), the system memory (604) comprising instructions that, when executed by the processing unit (602), cause the processing unit (602) to create:

a metadata processing module (204) that generates, monitors and stores metadata for a document on the computing device (102), the metadata including a document identifier, a next paragraph identifier and a revision identifier, the metadata processing module (204) generating a paragraph identifier range from the document identifier and from the next paragraph identifier, the paragraph identifier range including a first paragraph identifier and a last paragraph identifier, the first paragraph identifier having a value equal to the document identifier and the last paragraph identifier having a value equal to one less than the next paragraph identifier, wherein the metadata processing module (204) generates paragraph identifiers for each paragraph in the paragraph identifier range, each paragraph identifier in the paragraph identifier range being sequentially numbered; and

a metadata merge module (206) that merges the metadata for the document with metadata received from a server computer (106), wherein the metadata merge module (206) changes the document identifier to the document identifier included in the metadata received by the server computer (106) and wherein the metadata merge module (206) changes paragraph identifiers in the paragraph identifier range to paragraph identifiers included in the metadata received by the server computer (106).

14. The computing device of claim 13, wherein the metadata processing module stores metadata in a file on the server computer (106) that is separate from the document.

15. A computer-readable data storage medium comprising instructions that, when executed by a processing unit (602) of an electronic computing device (102), cause the processing unit (602) to:

obtain a document from a server computer (106), the document being requested by a user at the electronic computer device (102);

open the document;

in response to opening the document, determine whether the document includes paragraph identifiers;

when it is determined that the document does not include paragraph identifiers, assign a paragraph identifier to each paragraph in the document by:

randomly generate an identifier for the first paragraph of the document;

designate the identifier for the first paragraph of the document as a first document identifier;

sequentially assign identifiers to the remaining paragraphs in the document; and

designate a first next paragraph identifier having a value one greater than the identifier assigned to the last paragraph in the document;

receive a first message from the server computer (106) that at least one other user is authoring the document;

in response to receiving the first message from the server computer (106) that at least one other user is authoring the document, send a second message to the server computer (106), the second message including first metadata for the document, the first metadata including the first document identifier, the first next paragraph identifier, and a first revision identifier;

determine a first paragraph identifier range from the first document identifier and from the first next paragraph identifier, the first identifier in the first paragraph identifier range being the first document identifier and the last identifier in the first paragraph identifier range being one less than the first next paragraph identifier;

in response to sending the second message to the server computer (106), receive a third message from the server computer (106), the third message including second metadata for the document, the second metadata including a second document identifier, a second paragraph identifier range and a second revision identifier;

determine a second paragraph identifier range from the second document identifier and from the second next paragraph identifier, the first identifier in the second paragraph identifier range being the second document identifier and the last identifier in the second paragraph identifier range being one less than the second next paragraph identifier;

in response to receiving the third message from the server computer (106), determine whether the first metadata and the second metadata should be merged by:

determine whether the first revision identifier matches the second revision identifier;

when it is determined that the first revision identifier matches the second revision identifier, determine whether the first document identifier matches the second document identifier;

when it is determined that the first revision identifier matches the second revision identifier and when it is determined that the first document identifier does not match the second document identifier, determine whether the size of the first paragraph identifier range matches the size of the second paragraph identifier range;

when it is determined that the first revision identifier matches the second revision identifier, when it is determined that the first document identifier does not match the second document identifier and when it is determined that the size of the first paragraph identifier range matches the size of the second paragraph identifier range, determine that the first metadata and the second metadata should be merged; and

when it is determined that the first metadata and the second metadata should be merged, adjust paragraph identifiers in the document based on the second metadata by:

designate the second document identifier as the first document identifier, the second document identifier also being designated as the paragraph identifier for the first paragraph in the first paragraph identifier range;

assign paragraph identifiers in the second paragraph identifier range to corresponding paragraphs in the first paragraph identifier range;

determine whether a paragraph identifier in the document from a paragraph not included in the first paragraph identifier range matches a paragraph identifier in the second paragraph identifier range; and

when it is determined that a paragraph identifier in the document from a paragraph not included in the first paragraph identifier range matches a paragraph identifier in the second paragraph identifier range, assign the paragraph identifier in the document that matches a paragraph identifier in the second paragraph identifier range to a paragraph identifier in the first paragraph identifier range.

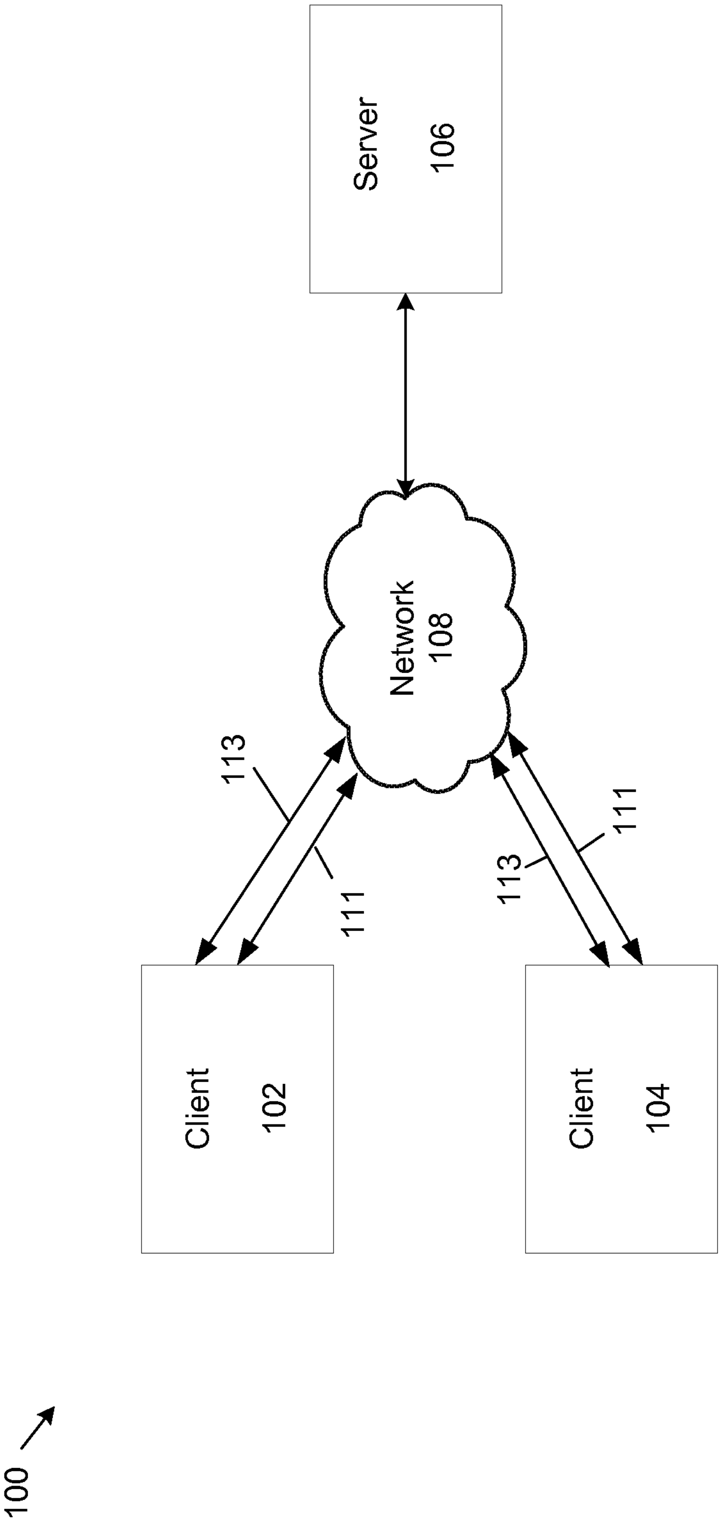
Server
106
Network
108
113
111
113
111
Client
102
Client
104
100

FIG. 1

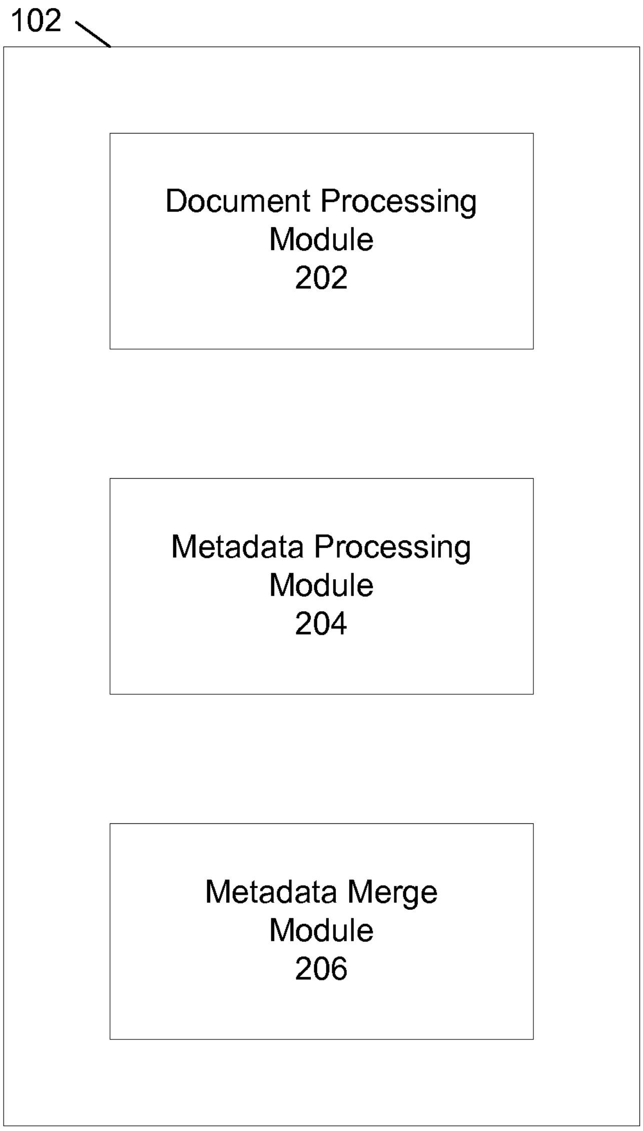
102
Document Processing Module 202
Metadata Processing Module 204
Metadata Merge Module 206

FIG. 2

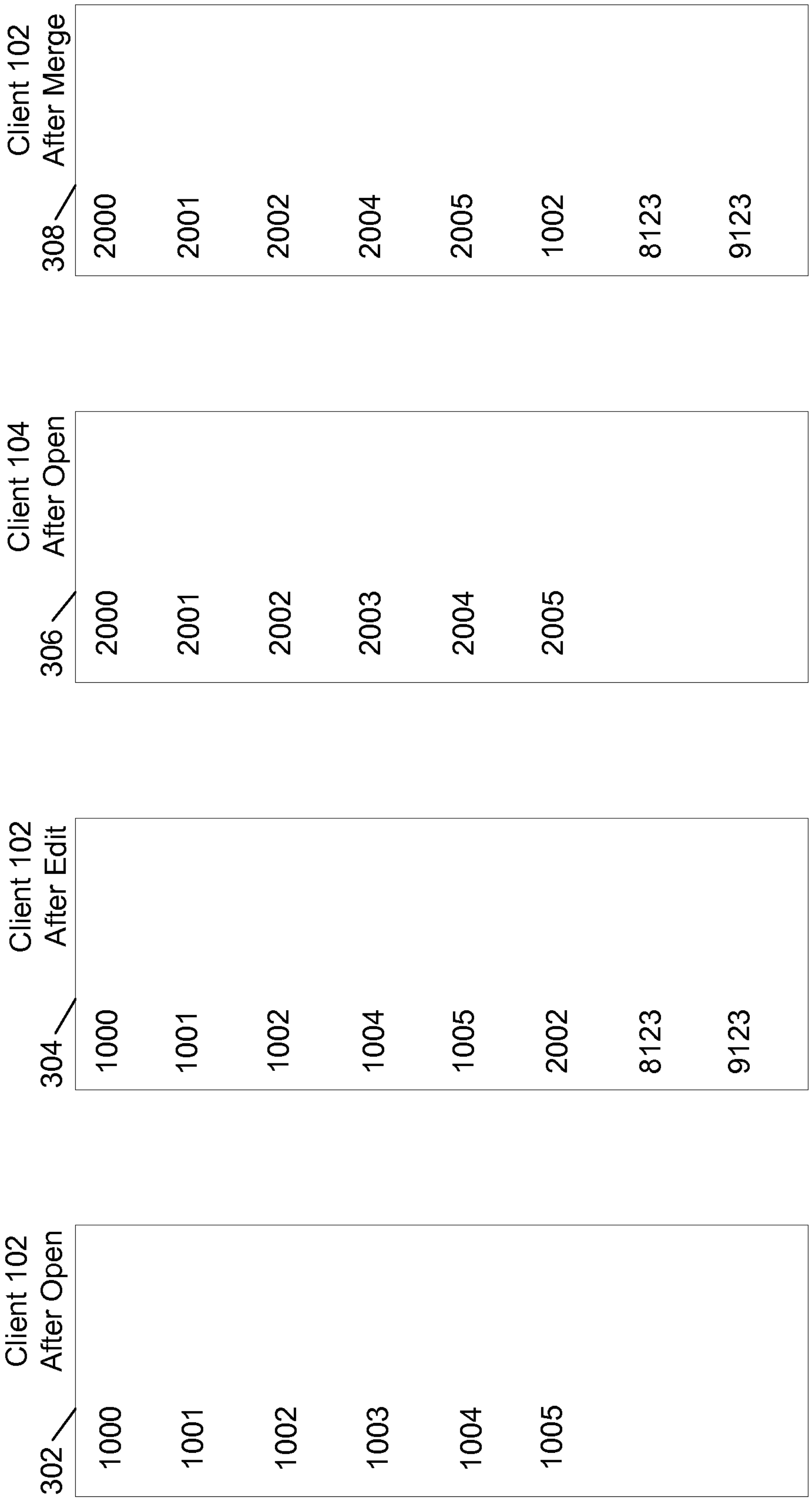
302
Client 102
After Open
1000
1001
1002
1003
1004
1005
304
Client 102
After Edit
1000
1001
1002
1004
1005
2002
8123
9123
306
Client 104
After Open
2000
2001
2002
2003
2004
2005
308
Client 102
After Merge
2000
2001
2002
2004
2005
1002
8123
9123

FIG. 3

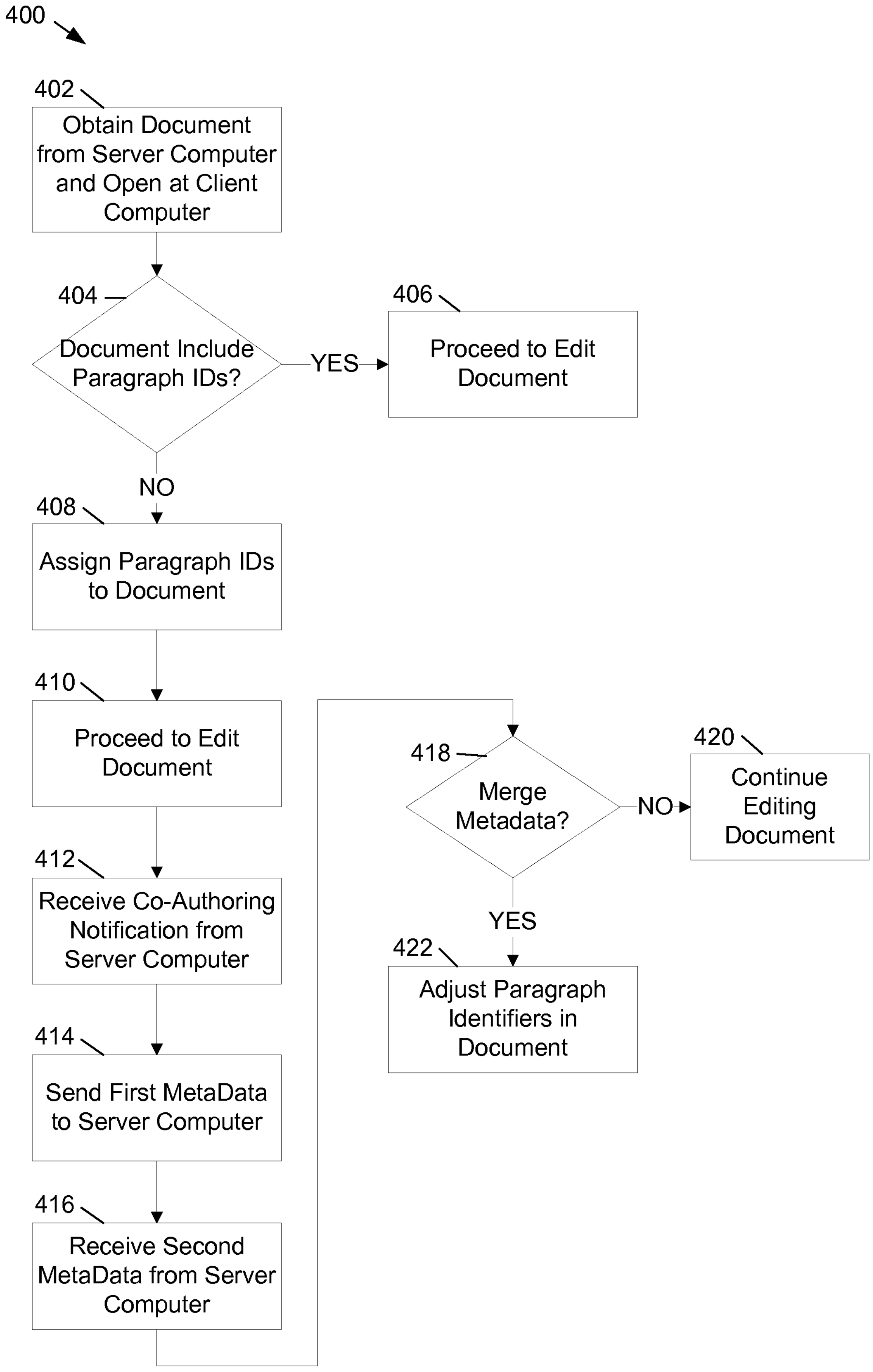
400
402
Obtain Document from Server Computer and Open at Client Computer
404
Document Include Paragraph IDs?
YES
406
Proceed to Edit Document
NO
408
Assign Paragraph IDs to Document
410
Proceed to Edit Document
412
Receive Co-Authoring Notification from Server Computer
414
Send First MetaData to Server Computer
416
Receive Second MetaData from Server Computer
418
Merge Metadata?
NO
420
Continue Editing Document
YES
422
Adjust Paragraph Identifiers in Document

FIG. 4

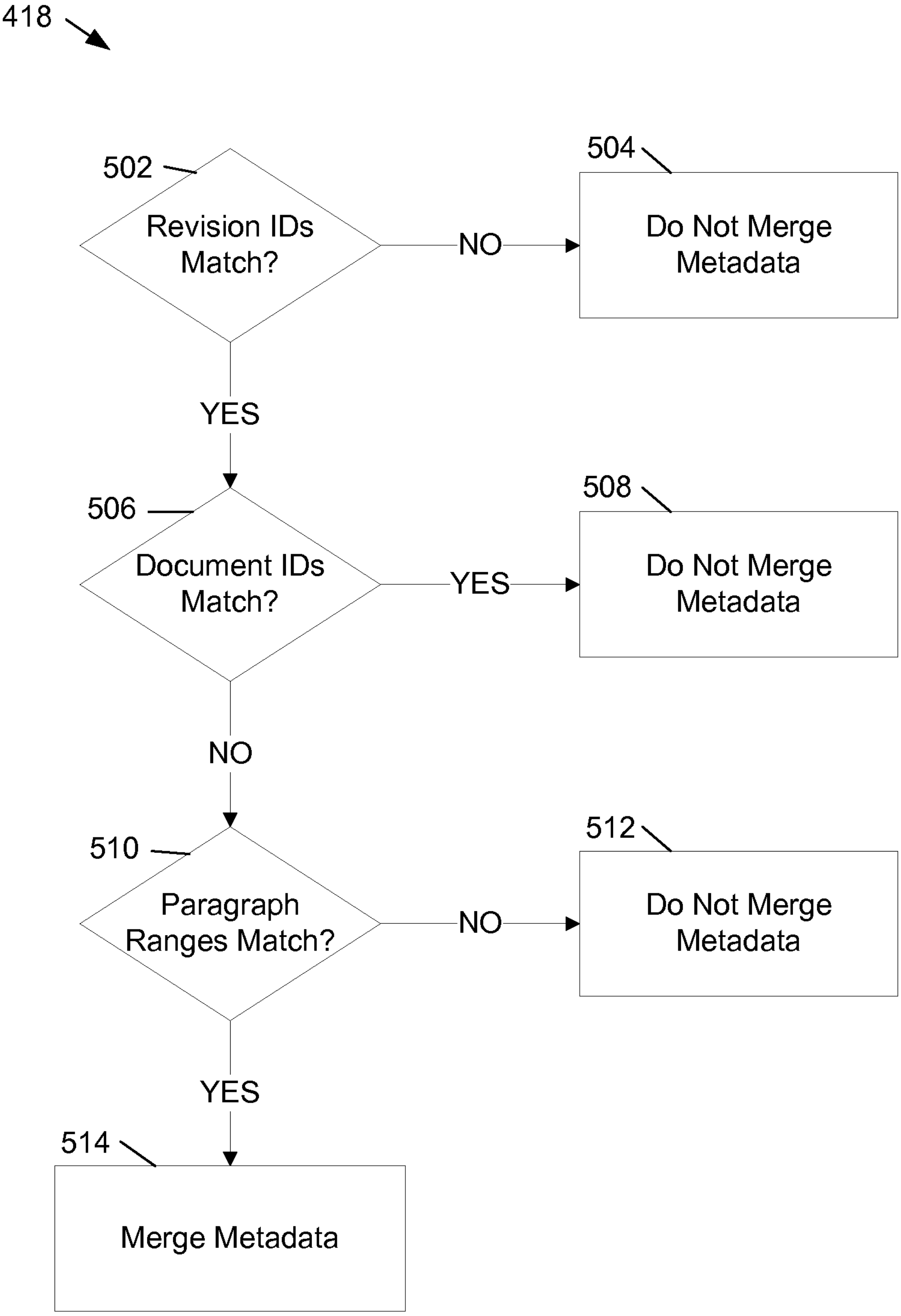
418
502
Revision IDs Match?
NO
504
Do Not Merge Metadata
YES
506
Document IDs Match?
YES
508
Do Not Merge Metadata
NO
510
Paragraph Ranges Match?
NO
512
Do Not Merge Metadata
YES
514
Merge Metadata

FIG. 5

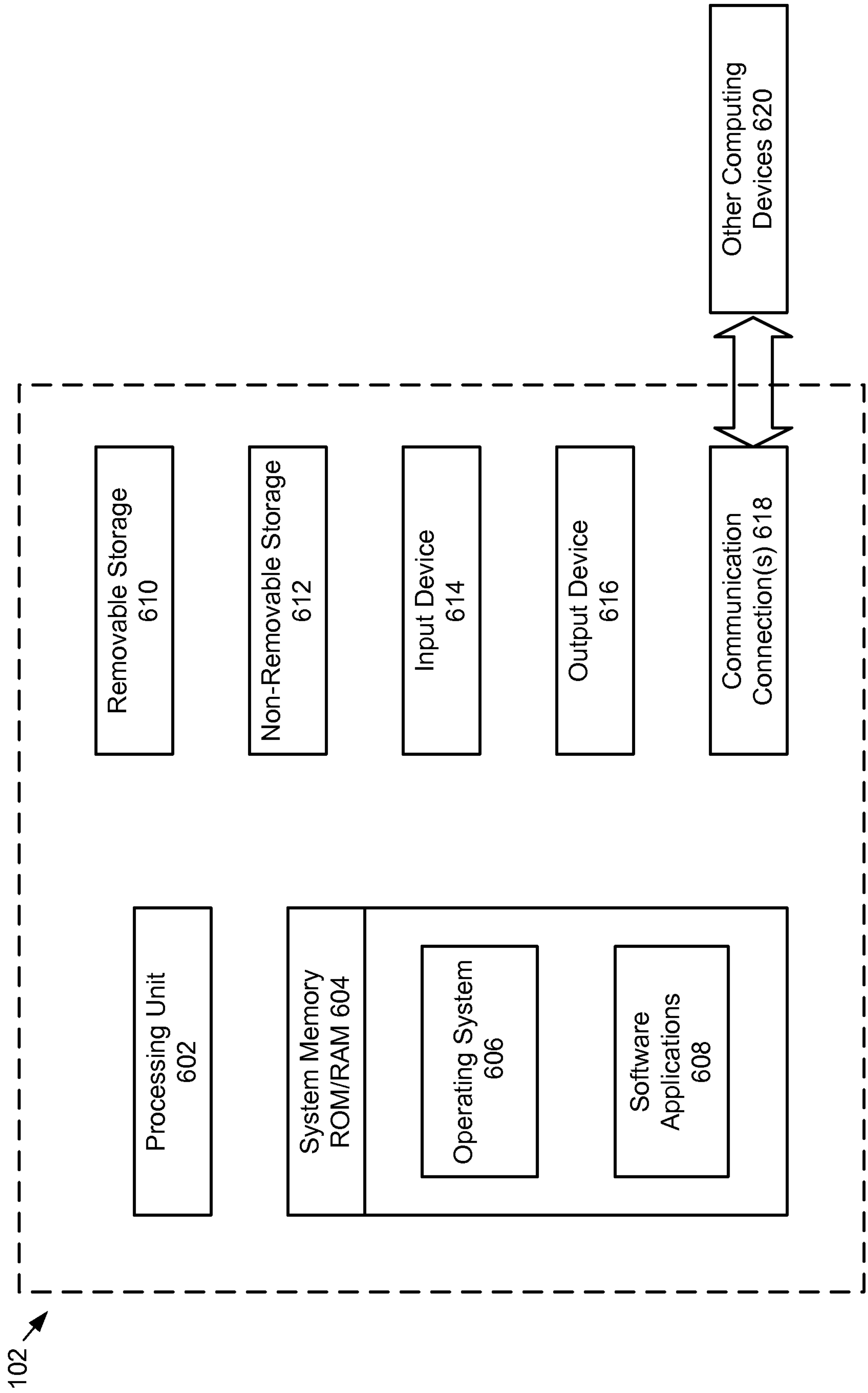
102
Processing Unit 602
System Memory ROM/RAM 604
Operating System 606
Software Applications 608
Removable Storage 610
Non-Removable Storage 612
Input Device 614
Output Device 616
Communication Connection(s) 618
Other Computing Devices 620

FIG. 6