[0093] In FIG. **20**, the second user adds a third data unit to the document and a third lock E**3** around the third data unit (e.g., in the memory of the authoring application). The third data unit contains the text "Goodbye World". The second authoring application generates (e.g., via a layer object) a working copy of the document including the third data unit and the third lock E**3**. The second authoring application also automatically provides (e.g., via a sync manager) the third lock E**3** to the storage device. However, the second authoring application does not provide the third data unit to the storage device.

[0094] Also in FIG. **20**, the first user releases the first lock E**1** after the first user indicates the first user has finished editing the first data unit. For example, the first user can provide instructions to store the document when the cursor of the first user is positioned within the second data unit. In one embodiment, the second data unit may be located remote from the first data unit. In the example shown, the finalization indication provided by the first user does not indicate the first user wishes to share content changes made by the first user. Accordingly, the first authoring application does not generate (e.g., via a layer object) an upload copy containing content changes for distribution to the storage device.

[0095] In FIG. **21**, the third lock E**3** has been stored in the master lock table of the master copy **1250** of the document on the storage device. The first user computing device subsequently polls (e.g., via a sync manager) the storage device to check for updates. The first user computing device obtained (e.g., via a sync manager) a new download copy of the document from the storage device based on the master copy **1250**. The new download copy contains the third lock E**3**. The first authoring application automatically instantiates (e.g., via a layer object) the third lock E**3** into the working copy of the first authoring application. As noted above, the third lock E**3** optionally may be displayed to the first user using the user interface of the first authoring application.

[0096] In FIG. **22**, the second user releases the third lock E**3** when the second user indicates the second user has finished editing the third data unit. The third lock E**3** is removed from the metadata table of the second user. In the example block diagram shown in FIG. **22**, the third lock E**3** is removed from the metadata tables corresponding to the working copy of the second authoring application.

[0097] In the example shown in FIG. **22**, the finalization indication provided by the second user indicates the second user wishes to share content changes with other users. For example, the second user may remove a cursor from the third data unit and then provide instructions to store the document. Accordingly, the second authoring application may generate an upload copy containing the content changes based on the working copy. In one embodiment, the upload copy also can be based on any content updates (e.g., in this case none) found in the download copy that had not yet been instantiated into the working copy. In the example shown, the upload copy includes the third data unit including the phrase "Goodbye World."

[0098] In FIG. **23**, the storage device obtains the third data unit from the upload copy of the second user computing device and merges the third data unit into the master copy **1250** of the storage device. The lock table associated with the master copy **1250** indicates the third lock E**3** has been released. As shown in FIG. **24**, upon a successful push to the storage device, the second authoring application copies (e.g., via a layer object) the upload copy to the download copy and the base copy of the second cache **1234**.

[0099] In FIG. **25**, the first user computing device obtains (e.g., via a sync manager) a new download copy of the master copy **1250** of the document. The first authoring application determines (e.g., via a layer object) the new download copy has been obtained and determines the new download copy includes content changes. The authoring application automatically merges any new locks (none in this case) into the working copy of the cache **1214**. The user interface of the first authoring application may indicate content updates are available for viewing.

[0100] In FIG. **26**, the first user chooses to instantiate the content updates available in the cache **1214** into the working copy of the document. Accordingly, the authoring application merges the download copy stored in the cache **1214** with the working copy of the first authoring application. The third data unit is added to the working copy of the first authoring application and is displayed to the first user. The third lock E**3** is released in the metadata table associated with the working copy of the first authoring application. The first user, therefore, may instruct the first authoring application to edit the third data unit.

[0101] Referring now to FIGS. **27-31**, the user experience during the coauthoring session is described in detail below. FIGS. **27-31** illustrate changes to an exemplary user interface **2000** of a first authoring application as a first user and a second user collaboratively author a document **2010**. The first user, Jane Doe, edits the document **2010** using the first authoring application and the second user, John Doe, edits the document **2010** using a different authoring application.

[0102] The document **2010** includes a first data unit **2012**, a second data unit **2014**, a third data unit **2016**, and a fourth data unit **2018**. The first user places a content lock **2022** around the first data unit **2012**. For example, in one embodiment, the first user may move her cursor **2015** to the first data unit **2012**. In another embodiment, the first user may provide instructions to perform one or more editing operations on the first data unit **2012**.

[0103] In FIG. **28**, the first user has added text to the first data unit **2012**. The lock on the first data unit inhibits the second user from editing the first data unit **2012**. In one embodiment, the first lock **2022** prevents the second user from editing the first data unit **2012**. In another embodiment, the first lock **2022** warns the second user against editing the first data unit **2012**, but does not bar editing by the second user. In yet another embodiment, the first lock **2022** merely indicates the first user has expressed an intention to edit the first data unit **2012**.

[0104] In FIG. **29**, the second user has placed a content lock **2024** about the third data unit **2016**. Because the second user has chosen to not share his changes, the second lock may be visible on the user interface **2000** of the first authoring application, but any changes made to the third data unit **2016** by the second user are not visible yet. In the example shown, the second lock **2024** is indicated by the user interface **2000** by displaying a bracket indicating the scope of the second lock **2024**. The second lock **2024** also can include a nametag indicating the owner of the second lock **2024**. In FIG. **29**, the nametag includes text identifying the second user and an icon. The icon can further identify a user by color, symbol, or other graphic.

[0105] In FIG. **30**, the first authoring application receives content updates for the document **2010**. The second user has