mentation, the system can use the key file method as described above. In an alternative implementation, the system can refer to a single file that contains information relating to the update components already installed.

[0042] The system displays the items of the second list aligned with the items of the first list (step **220**). The names of the installed update components are displayed in a similar format as the list of available update components. In one implementation, the order of the second list will be the same order as that of the first list, so that identical entries on each list are aligned. This can be seen in **FIG. 3**, where available update component **330** in the first list of available update components **305** is displayed across from the same installed component **335** in the second list. In the second list of installed update components **310**, the names of the installed updates are listed in hierarchical format, to match the display order in the first list of available update components **305**. In another implementation, update component information in addition to or instead of the component name can also be displayed in the second list.

[0043] In order for the second list to remain aligned with the first list, blank spaces appear when an available update that appears on the first list is not installed. In **FIG. 3**, blank space **325** indicates that the available "9.0.1 Update" component listed in the first list of available update components **305** is not installed. In addition to keeping the first list and the second list aligned, the blank space **325** also enables the user to easily determine what update components have not yet been installed. In an alternative embodiment, an alternative placeholder can be used, e.g., a brief message indicating that the component has not yet been installed.

[0044] The system allows the user to select one or more components to install (step **225**). The user can select any available update components, including components that have already been installed, as well as components that have not yet been installed. A user may wish to select a previously installed component for a variety of reasons. For example, the component may have become corrupted or damaged. In one implementation, the user selects each desired update component by selecting a check box **340** adjacent to the name of the update component in the user interface **300**.

[0045] The system requests and receives the selected update components (step **230**). In one implementation, the location of each update component is indicated in the first list of available update components received (step **205**). Update components can be requested and received from a variety of sources; for example, a request for the update component can be sent over a computer network, e.g., the Internet or a local area network (LAN), and the update component can be received over the same computer network. Alternatively, the update component can be received over a different computer network. The location of the update component can be provided by means of a uniform resource locator (URL) or by means of a web service call. Alternatively, the components can be made available by means of removable media, such as a floppy disk, CD-ROM, or DVD.

[0046] The system installs selected update components (step **235**). In one implementation, this is performed automatically by the system. After the user selects the desired components in step **225**, the system automatically installs the components without any further intervention by the user. In an alternative implementation, the user is prompted to install the components manually.

[0047] **FIG. 4** shows an example user interface **400** displayed to the user when components are ready to be installed. The user can confirm the components to be installed by ensuring that the check box **405** is selected.

[0048] The system can operate as an automated system that checks for updates periodically, depending on the needs of the user. For example, a user might wish to have the invention check for updates on a weekly basis. In this case, at a specific time each week, the list of available update components would be received, and if any new update components are available, the user will be notified. Alternatively, if there are any update components available that are not installed on the user's computer, the user will be notified. The user can choose to complete the update process as described above. The invention can also be applied to itself. For example, the update software component can check to see if there are any update components for itself, and install them if desired. In another alternative, the user might wish to check for updates at a regular interval, e.g., once a month, upon starting one of the installed programs. In this instance, after starting one of the installed programs, if the requested time period between checking for updates has elapsed, a check for updates occurs. In this manner, a separately running program to check for updates is not required, requiring the use of fewer system resources, such as memory and CPU usage.

[0049] The user can also elect to search for update components at any time. In one implementation, the user can launch any program supported by the updater, and from within the software run the updating process. For example, in all Adobe® software applications that feature updates, a user can manually initiate an update request by selecting Updates from the Help drop-down menu.

[0050] The invention can also be implemented across a network of computers, so that update components can be installed from a remote location. For example, a network administrator can install update components on multiple computers from a single location. The network administrator can install update components on an as-needed basis for each computer on the network, or can have the invention check for new update components on a regular basis, for example, every day or every week. Any new updates can then be installed automatically on each computer on the network.

[0051] The invention can also be implemented to distribute a variety of components in addition to update components. These components can include marketing announcements, new products, free products, trial products, or other similar items, that the user can download if desired. In such an instance, the manifest file that is distributed contains information that is associated with the components being distributed. The user can then view the available components and the installed components in the dual-column display format discussed above, and can select any desired components for download and installation.

[0052] Further, the manifest files and update components can be kept on a local area network (LAN), for example on a server accessible from within the LAN. Client computers can request manifest files from the server on the LAN, instead of or in addition to requesting manifest files on the Internet or through some other means. In addition, the update components can also be available from the server. This implementation also has the added benefit of reducing Internet bandwidth requirements, as each computer on the LAN can access manifest files and update components locally.