DISPLAYING A LIST OF FILE ATTACHMENTS ASSOCIATED WITH A MESSAGE THREAD

BACKGROUND

[0001] The development of reports, presentations, and other documents often involves collaboration between multiple, remote individuals within an organization or group. However, even in corporate environments that have access to collaboration software or other file sharing technologies, individuals often use email to collaborate, utilizing file attachments to share documents with each other or to distribute versions of documents for review and editing.

[0002] For example, one person may send an initial email message containing an attached document to several individuals in the organization. Each of these individuals may in turn revise the documents, often renaming the document in such a way to reflect that modifications have been made. The modified document may then be returned to the sender in a reply message, or forwarded to others in the organization. The messages sent back and forth in response to the initial message together make up a conversation or message thread. The message thread may eventually include several file attachments representing different versions of the original document, referred to herein as the root file.

[0003] Users accessing their emails through traditional email applications may be provided with a mechanism to view all the messages associated with a message thread, either in date order or in a hierarchical display indicating the flow of messages among the individuals involved in the message thread. The display of messages associated with the message thread may also provide an indication of the messages that include a file attachment.

[0004] However, there is currently no easy way to aggregate together all of the different file attachments that have been attached or modified by the individuals involved in the conversation or message thread. An email user typically accesses each message in the message thread that has an attachment manually in order to identify the root files associated with the message thread. Further, users typically determine manually the most recent version of each root file by reviewing the varying names and last modified dates of each file attachment related to the root file.

[0005] It is with respect to these considerations and others that the disclosure made herein is presented.

SUMMARY

[0006] Technologies are described herein for displaying a list of file attachments associated with a message thread in an email application. The messages associated with the message thread are identified, along with any file attachments attached to those messages. An aggregated attachment list is generated containing an entry for each file attachment associated with the message thread. The aggregated attachment list associated with the message thread is then displayed in a window of the email application.

[0007] It should be appreciated that the above-described subject matter may be implemented as a computer-controlled apparatus, a computer process, a computing system, or as an article of manufacture such as a computer-readable medium. These and various other features will be apparent from a reading of the following Detailed Description and a review of the associated drawings.

[0008] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended that this Summary be used to limit the scope of the claimed subject matter. Furthermore, the claimed subject matter is not limited to implementations that solve any or all disadvantages noted in any part of this disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIGURE 1 is a block diagram showing aspects of illustrative operating environments and several software components provided by the embodiments presented herein;

[0010] FIGURES 2-3 are screen diagrams showing exemplary user interfaces for displaying a list of file attachments associated with a message thread, according to embodiments described herein;

[0011] FIGURE 4 is a flow diagram showing methods for displaying a list of file attachments associated with a message thread, as provided in the embodiments described herein;

[0012] FIGURE 5 is a flow diagram showing methods for identifying the root files related to the file attachments associated with a message thread, as provided in the embodiments described herein; and

[0013] FIGURE 6 is a block diagram showing illustrative computer hardware and software architectures for a computing system capable of implementing aspects of the embodiments presented herein.

DETAILED DESCRIPTION

[0014] The following detailed description is directed to technologies for displaying an aggregated attachment list in an email application that contains a list of file attachments associated with a message thread or conversation. The aggregated attachment list allows a user to view and interact with all the file attachments or root files associated with the message thread in a single location. By providing the aggregated attachment list, the user no longer needs to manually access each message in the message thread to identify the root files associated with the message thread. Nor will the user have to manually determine the most recent version of each root file by reviewing the varying names and last modified dates of each file attachment related to the root file.

[0015] While the subject matter described herein is presented in the general context of program modules that execute in conjunction with the execution of an operating system and application programs on a computer system, those skilled in the art will recognize that other implementations may be performed in combination with other types of program modules. Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the subject matter described herein may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like.

[0016] In the following detailed description, references are made to the accompanying drawings that form a part hereof, and which show by way of illustration specific embodiments or examples. Referring now to the drawings, in which like numerals represent like elements through the several figures, aspects of a methodology for displaying a list of file attachments associated with a message thread will be presented.

[0017] Turning now to FIGURE 1, details will be provided regarding an illustrative operating environment and several software components provided by the embodiments presented herein. FIGURE 1 illustrates an exemplary system 100 for displaying an aggregated attachment list in an email application. The system 100 includes

a workstation 102 that is connected to a display 104 allowing the workstation 102 to display a user interface (UI) to a user 106. It will be appreciated that the user 106 may interact with the workstation 102 through any number of input and output devices, such as a computer monitor, printer, keyboard, mouse, trackball, mouse pad, stylus, or touch-screen that may be directly connected to the workstation 102 or remotely connected via a network. It will be further appreciated that the workstation 102 illustrated in FIGURE 1 may represent any computing device with which a user 106 may interact, including a desktop computer, notebook, laptop, tablet computer, ultra-mobile PC, server, or handheld device.

[0018] An email application 108 application executes on the workstation 102 that allows the user 106 to access and manage messages and other information. For example, the email application 108 may be a personal information manager (PIM) that provide access to email, voicemail, personal contacts, organizational contacts, calendar events, and other personal or business data and processes. In one embodiment, the email application 108 is the MICROSOFT® OFFICE OUTLOOK® messaging and collaboration client from MICROSOFT CORPORATION of Redmond, Washington. It should be appreciated, however, that the embodiments presented herein may be utilized with email application programs from other vendors, including but not limited to, IBM® LOTUS NOTES® software from IBM CORP. of Armonk, New York, NOVELL® GROUPWISE® software from NOVELL, INC. of Waltham, Massachusetts, or GMAIL™ from GOOGLE INC. of Mountain View, California.

[0019] The email application 108 is connected through a network 110 to a messaging server 112 executing on a remote server platform 114. The messaging server 112 provides messaging and other information services to the email application 108. In one embodiment, the messaging server 112 is the MICROSOFT® EXCHANGE SERVER messaging server from MICROSOFT CORPORATION. As will be appreciated by one skilled in the art, the messaging server 112 may be any messaging server known in the art capable of providing the messaging services described herein to the email application 108, such as IBM® LOTUS DOMINO® software from IBM CORP. or NOVELL® GROUPWISE® software from NOVELL, INC.

[0020] According to example embodiments, the network 110 comprises the Internet. However, it will appreciated that the network 110 may comprise a LAN, WAN, or other type of network suitable to allow the email application 108 located on the workstation 102 to communicate with the messaging server 112 executing on the server

platform 114. It will be further appreciated that many more networks and network connections may be utilized than illustrated in FIGURE 1.

[0021] The messaging server 112 may be connected to a message store 116. The message store 116 may store messages 118 and other data items for access by the email application 108. A message 118 may represent an email message, text message, fax, voice-mail, or any other communication. The message 118 may contain a subject 120. In example embodiments, the message 118 may also contain a conversation ID 122 that is used to group messages into conversations or message threads, as will be discussed in more detail below in regard to FIGURE 4. According to further embodiments described herein, the message 118 may also include a number of file attachments 124. A file attachment 124 is a document or file attached to the message 118. The file attachment 124 may have a file name 126, a modified date 128 indicating the date the file was last modified, a file size 130 indicating the size of the file in bytes, and an author ID 132 indicating the author of the document or file. As will be appreciated by those skilled in the art, the message store 116 may be located on the server platform 114 or on a remote storage device accessible by the messaging server 112. The message store 116 may also contain other data items such as calendar events, contacts, tasks, notes, and other personal or business data beyond that illustrated in FIGURE 1.

[0022] In further embodiments, the email application 108 may include a local storage 134 where the email application 108 caches data regarding messages 118 and other data items. A message 118 may be cached in the local storage 134 upon the first access of the message 118 by the email application 108, or all available messages 118 may be moved or copied from the message store 116 to the local storage 134 on a scheduled basis. For example, the email application 108 may access the messaging server 112 via the Post Office Protocol version 3 (POP3) protocol every hour to download the available messages 118 from the message store 116 and store them in the local storage 134. It will be appreciated that many other methods and protocols for retrieving messages from a messaging server and locally storing the messages may be utilized by the email application 108.

[0023] The email application 108 may also include an attachment aggregation module 136 that performs the operations of displaying the aggregated attachment list, which will be described in detail below. In example embodiments, the functionality of the attachment aggregation module 136 is provided by the email application 108. It will be appreciated, however, that the attachment aggregation module 136 may be provided as an

add-in module for the email application 108. The add-in module may utilize an application programming interface (API) provided by the email application 108 to access the data from the messaging server 112 or the local storage 134 and to display the aggregated attachment list as part of the UI of the email application 108.

[0024] FIGURES 2–3 illustrate exemplary UIs that may be generated by the attachment aggregation module 136 and the email application 108 for displaying an aggregated attachment list associated with a message thread, according to embodiments. In particular, FIGURE 2 illustrates a UI 200 generated by the attachment aggregation module 136 and email application 108 in conjunction with the display of data regarding a message 118 associated with a message thread. The UI 200 includes of a window 202 displayed by the email application 108 that contains a group of messages 118A-118I associated with a message thread 204 or conversation.

[0025] The messages 118A-118I associated with the particular message thread 204 may be identified based on a shared conversation ID 122, same or similar subjects 120, or some other method of determining the relationships among messages, as will be described in more detail below in regard to FIGURE 4. For example, the message thread 204 illustrated in FIGURE 2 contains several associated messages 118A-118I grouped under a heading 206 of "New Project Plan Document…." The presence of one or more file attachments 124 for a particular message 118A may be indicated by an attachment indicator 208 image or icon. Another attachment indicator 208 may be displayed next to the heading 206 of the message thread 204 to indicate that the message thread 204 includes at least one message 118A with a file attachment 124. When a particular message 118F is selected by the user 106 from the message thread 204, as indicated by the selection box 210 in FIGURE 2, information regarding the selected message 118F is displayed by the email application 108 in a message detail pane 214 of the window 202.

[0026] According to embodiments presented herein, when the message detail pane 214 is displayed for a message 118F associated with a message thread 204 that includes at least one file attachment 124, the attachment aggregation module 136 displays an aggregated attachment list 216 in the window 202 in proximity to the message detail pane 214. In example embodiments, the aggregated attachment list 216 may contain list entries 218A-218C for the file attachments 124 attached to the messages 118A-118I associated with the message thread 204 of which the selected message 118F is a part. The list entries 218A-218C may contain a file-type icon 220 indicating the file type of the corresponding

file attachment 124, the name 222 of the file, the author 224 of the file, and the size 226 of the file in bytes.

[0027] As described above, the file attachments 124 associated with the message thread may represent several different versions of a few, root files. In further embodiments, the aggregated attachment list 216 may contain list entries 218A-218C for any root file identified from the file attachments 124 associated with the message thread 204, as will be described in more detail below in regard to FIGURE 5. The file-type icon 220, name 222, author 224, and file size 226 displayed for the list entries 218A-218C represents the latest version of the corresponding root file. In addition, list entries 218A-218B for corresponding root files having multiple versions identified in the file attachments 124 may also include an indication of the number of versions 228 of the root file represented in the file attachments 124.

[0028] The latest version of a root file and the number of versions available may be determined by the attachment aggregation module 136, as will be described below in regard to FIGURE 5. The number of versions 228 may be displayed as a hyperlink that allows the user 106 to access the various versions of the root file when the user 106 selects the hyperlink using a mouse, for example. It should be appreciated that, although the use of a mouse is described in the examples presented herein, any suitable user input devices may be utilized to allow users to select or interact with UI controls, including, but not limited to, a keyboard, trackball, mouse pad, stylus, or touch-screen that may be connected to the workstation 102.

[0029] According to example embodiments, the attachment aggregation module 136 may provide the same UI functionality for the various list entries 218A-218C displayed in the aggregated attachment list 216 as provided for any other representation of a file in the email application 108. For example, a list entry 218B may be selected by a user 106 with a mouse, dragged from the aggregated attachment list 216, and dropped into a file folder or other file location presented in the UI. If a list entry 218B is dragged from the aggregated attachment list 216 and dropped into a file location, the latest version of the root file corresponding to the list entry 218B may be copied to the file location.

[0030] The attachment aggregation module 136 also provides for files to be dragged and dropped into the aggregated attachment list 216 through the UI. If a file is dropped into the aggregated attachment list 216 from another location, the attachment aggregation module 136 may create a new message 118 associated with the message thread 204, attach the file as a file attachment 124, and store the message 118 in the

message store 116 or the local storage 134. Additionally, the attachment aggregation module 136 may forward the message 118 to some or all of the individual senders and recipients of messages 118A-118I associated with the message thread 204.

[0031] In another embodiment, the attachment aggregation module 136 may provide further functionality for managing the root files or file attachments 124 corresponding to the list entries 218A-218C displayed in the aggregated attachment list 216. As illustrated in the UI 300 depicted in FIGURE 3, this additional functionality may be accessed from a context menu 302 displayed by the attachment aggregation module 136 when a user 106 hovers a mouse cursor 304 over the aggregated attachment list 216 and clicks the right mouse button, for example. The context menu 302 may contain menu items that allow the user 106 to perform actions on the entire group of attachments, such as:

- "Save All Attachments" that would save all file attachments or root files associated with the message thread 204 to a single location;
- "Forward All Attachments" that would allow the user 106 to create a new message 118 with all of the root files attached;
- "Remove Previous Versions" that removes all versions of all root files except for the latest version of each from the message store 116 or local storage 134; and
- "Filter by Sender" that allows the user to view just the file attachments 124 from an individual sender.

It will be appreciated that the context menu 302 may contain other menu items that allow the user 106 to act upon all file attachments 124 or an individual file attachment 124 or root file associated with the message thread 204.

[0032] While the aggregated attachment list 216 is illustrated in FIGURES 2-3 as being displayed in conjunction with the data regarding a message 118F selected from a message thread 204, it will be appreciated that the attachment aggregation module 136 may generate an aggregated attachment list 216 in conjunction with the display of any message in the message store 116 that is associated with a message thread 204, regardless of the context of its selection. Further, many variations in the location of the aggregated attachment list 216 in the window 202 and the layout of the file-type icon 220, name 222, author 224, file size 226, and number of versions 228 displayed in the aggregated attachment list 216 will be apparent to those of ordinary skill in the art, and this

application is intended to cover any such variations beyond those illustrated in FIGURES 2-3.

[0033] Referring now to FIGURES 4 - 5, additional details will be provided regarding the embodiments presented herein. In particular, FIGURE 4 illustrates routines, denoted generally at 400, for displaying an aggregated attachment list associated with a message thread. It should be appreciated that the logical operations described herein are implemented (1) as a sequence of computer implemented acts or program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance and other requirements of the computing system. Accordingly, the logical operations described herein are referred to variously as states operations, structural devices, acts, or modules. These operations, structural devices, acts, and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof. It should also be appreciated that more or fewer operations may be performed than shown in the figures and described herein. These operations may also be performed in a different order than those described herein.

[0034] The routines 400 begin at operation 402, where the attachment aggregation module 136 identifies the messages 118A-118I associated with the current message thread 204. As described above, this may be done by identifying all messages 118 in the message store 116 or local storage 134 that share a common conversation ID 122, according to example embodiments. The conversation ID 122 may be a globally unique identifier (GUID) or a string that identifies a conversation or message thread to which the message 118 is associated. The conversation ID may be generated for the first message 118A in a message thread. Subsequent messages 118B-118I in the message thread 204, such as replies or forwarded messages, may have the same conversation ID 122 as the original message 118A. This allows subsequent messages 118B-118I to retain their association with the message thread 204, even if other data regarding the message 118, such as the subject 120, is changed.

[0035] In further embodiments, the attachment aggregation module 136 may utilize the subject 120 of a given message to determine the messages 118A-118I associated with the message thread 204. In example email applications 108, when a user 106 replies to or forwards a message 118A, the subsequent message 118B-118I may retain the same subject 120 as the original message 118A, except for a prefix, such as "Re:" or "FW:," appended to the subject 120 to indicate the action taken by the user 106 upon the

previous message. In determining the messages 118A-118I associated with the message thread 204, the attachment aggregation module 136 may identify all messages 118 in the message store 116 or local storage 134 that share a similar subject 120, ignoring any prefixes in the subject 120. It will be appreciated that other methods of identifying messages associated with a message thread may be used by the attachment aggregation module 136 in operation 402.

[0036] From operation 402, the routines proceed to operation 404, where the attachment aggregation module 136 identifies any file attachments 124 of the messages 118A-118I associated with the message thread 204. The attachment aggregation module 136 may access the messages 118A-118I identified in operation 402 to determine the file attachments 124 associated with the message thread 204. Once the file attachments 124 associated with the message thread 204 have been identified, the routines 400 may proceed from operation 404 to operation 406, where the attachment aggregation module 136 identifies the root files for the file attachments 124, as will be described in more detail below in regard to FIGURE 5.

[0037] The routines 400 then proceed from operation 406 to operation 408 where the attachment aggregation module 136 displays the aggregated attachment list 216 in the window 202 of the email application 108. A list entry 218A-218C may be created in the aggregated attachment list 216 for the root files associated with the message thread 204, as identified by the attachment aggregation module 136 in operation 406. The attachment aggregation module 136 uses the file name 126, file size 130 and author ID 132 retrieved from the file attachment 124 determined to be the latest version of the corresponding root file to render the file-type icon 220, name 222, author 224, and file size 226 for the list entries 218A-218C in the aggregated attachment list 216. In addition, the number of versions 228 of each root file determined in operation 406 is displayed in the corresponding list entry 218A-218C.

[0038] FIGURE 5 illustrates routines, denoted generally at 500, that provide further aspects of the operation 406 described above for identifying the root files associated with the message thread 204 from the file attachments 124. The routines 500 begin at operation 502 where the attachment aggregation module 136 accesses the file attachments 124 identified in operation 404 above as being associated with the message thread 204. For a given file attachment, the routines 500 proceed to operation 504 where the attachment aggregation module 136 determines if the file attachment 124 is related to a root file already identified for the message thread 204.

[0039] According to example embodiments, the attachment aggregation module 136 may use the file name 126 of the file attachments 124 to determine if two file attachments 124 are related to the same root file or document. If the file names 126 of two file attachments 124 are the same, then the attachment aggregation module 136 may determine that the file attachments 124 are related to the same root file.

[0040] However, when a user 106 replies to or forwards a message 118 containing a modified file attachment 124, the user 106 may rename the file attachment 124 to indicate that the document or file has been revised. For example, a user may receive a message 118A with a file attachment named "Project X Project Plan.xls." The user then subsequently modifies the document attached to the message 118A, and replies with a new message 118G attaching the modified document as a file attachment 124 with the name "Project X Project Plan (BillJ revisions 9-15).xls." In order to determine that the file attachments 124 attached to the two messages 118A, 118G are versions of the same root file, the attachment aggregation module 136 may analyze the file name 126 of each file attachment 124 by applying common syntactical rules or methods for such renaming of files. It will be appreciated that many methods of syntactical logic may be applied to determine file attachments related to the same root file and versioning of root files based on the file names.

[0041] In further embodiments, the attachment aggregation module 136 may use other attributes of the file attachments 124 to determine if they are related to the same root file, such as a document title or a description extracted from the contents of the file attachment 124. It will be appreciated that many other methods may be employed by the attachment aggregation module 136 to compare the file attachments, including a comparison of the contents of the file attachments.

[0042] At operation 506, if the attachment aggregation module 136 determines that the file attachment 124 is related to an already identified root file, then the routines 500 may proceed to operation 508. In operation 508, the attachment aggregation module 136 may determine the version level of the file attachment 124, based on a comparison with the identified root file. By determining a version level of each file attachment 124 related to the same root file, the attachment aggregation module 136 may determine the file attachment 124 that represents the latest version of the root file, as well as the number of versions of the root file available for display in the aggregated attachment list 216. In example embodiments, the attachment aggregation module 136 may determine the version

levels of the file attachments related to the same root file by comparing the last modified date 128 of the file attachments 124.

[0043] If, at operation 506, the attachment aggregation module 136 determines that the file attachment 124 is not related to an already identified root file, the routines 500 may proceed to operation 510, where a new root file is identified for the message thread 204 from the file attachment 124. Next, at operation 512, if there are more file attachments to be processed, the routines 500 may return to operation 504, where the attachment aggregation module 136 repeats the routine 500 for the next file attachment 124. If no more file attachments 124 are left to process, the routines 500 may proceed from operation 512 to operation 514 where the routine 500 ends.

[0044] FIGURE 6 shows illustrates example computer architectures for computers 600 capable of executing the software components described herein for displaying a list of file attachments associated with a message thread in the manner presented above. The computer architecture shown in FIGURE 6 illustrates a conventional desktop, laptop, or server computer and may be utilized to execute any aspects of the software components presented herein described as executing on the workstation 102 or server platform 114.

[0045] The computer architecture shown in FIGURE 6 includes a central processing unit 602 (CPU), a system memory 608, including a random access memory 614 (RAM) and a read-only memory 616 (ROM), and a system bus 604 that couples the memory to the CPU 602. A basic input/output system containing the basic routines that help to transfer information between elements within the computer 600, such as during startup, is stored in the ROM 616. The computer 600 also includes a mass storage device 610 for storing an operating system 618, application programs, and other program modules, which are described in greater detail herein.

[0046] The mass storage device 610 is connected to the CPU 602 through a mass storage controller (not shown) connected to the bus 604. The mass storage device 610 and its associated computer-readable media provide non-volatile storage for the computer 600. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, it should be appreciated by those skilled in the art that computer-readable media can be any available computer storage media that can be accessed by the computer 600.

[0047] By way of example, and not limitation, computer-readable media may include volatile and non-volatile, 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. For example, computer-readable media includes, but is not limited to, RAM, ROM, EPROM, EEPROM, flash memory or other solid state memory technology, CD-ROM, digital versatile disks (DVD), HD-DVD, BLU-RAY, or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to store the desired information and that can be accessed by the computer 600.

[0048] According to various embodiments, the computer 600 may operate in a networked environment using logical connections to remote computers through a network such as the network 110. The computer 600 may connect to the network 110 through a network interface unit 606 connected to the bus 604. It should be appreciated that the network interface unit 606 may also be utilized to connect to other types of networks and remote computer systems. The computer 600 may also include an input/output controller 612 for receiving and processing input from a number of other devices, including a keyboard, mouse, or electronic stylus (not shown in FIGURE 6). Similarly, an input/output controller may provide output to a display 104, a printer, or other type of output device.

[0049] As mentioned briefly above, a number of program modules and data files may be stored in the mass storage device 610 and RAM 614 of the computer 600, including an operating system 618 suitable for controlling the operation of a networked desktop, laptop, or server computer. The mass storage device 610 and RAM 614 may also store one or more program modules. In particular, the mass storage device 610 and the RAM 614 may store the email application 108 or the messaging server 112, each of which was described in detail above in regard to FIGURE 1. The mass storage device 610 and the RAM 614 may also store other types of program modules or data.

[0050] Based on the foregoing, it should be appreciated that technologies for displaying a list of file attachments associated with a message thread in an email application are provided herein. Although the subject matter presented herein has been described in language specific to computer structural features, methodological acts, and computer readable media, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features, acts, or media described herein. Rather, the specific features, acts, and mediums are disclosed as example forms of implementing the claims.

[0051] The subject matter described above is provided by way of illustration only and should not be construed as limiting. Various modifications and changes may be made

to the subject matter described herein without following the example embodiments and applications illustrated and described, and without departing from the true spirit and scope of the present invention, which is set forth in the following claims.

What is claimed is:

1. A method for displaying an aggregated attachment list (216) associated with a message thread (204) in an email application (108), the method comprising:

identifying one or more messages (118) associated with the message thread (204);

identifying one or more file attachments (124) attached to the one or more messages (118);

generating the aggregated attachment list (216) to include respective list entries (218A-218C) for the one or more file attachments (124); and

displaying the aggregated attachment list (216) in a window (202) of the email application (108).

2. The method of claim 1, wherein the list entries respectively comprise one or more of a file-type icon, a file name, a file size, and an identification of an author for the one or more file attachments.

3. The method of claim 1, wherein identifying the one or more messages associated with the message thread comprises identifying one or more messages having a same conversation ID.

4. The method of claim 1, wherein identifying the one or more messages associated with the message thread comprises identifying one or more messages having a same or similar subject.

5. The method of claim 1 further comprising:

upon identifying the one or more file attachments, identifying respective root files related to the one or more file attachments;

determining respective version levels for the one or more file attachments based upon the root file related to the file attachment; and

generating the aggregated attachment list to include respective list entries for the root files identified.

6. The method of claim 5 wherein the list entries respectively comprise one or more of a file-type icon, a file name, a file size, and an identification of an author of the file attachment related to the root files having a greatest version level.

7. The method of claim 5 wherein the list entries respectively further comprise an indication of a number of file attachments related to the root files.

8. The method of claim 5, wherein identifying the respective root files related to the one or more file attachments comprises identifying file attachments having a same or similar file name as related to one root file.

9. The method of claim 5, wherein determining respective version levels for the one or more file attachments based upon the root file related to the file attachment comprises comparing a respective modified date for the one or more file attachments related to the root file.

10. A computer readable storage medium having computer executable instructions stored thereon that, when executed by a computer, cause the computer to:

identify one or more messages (118) associated with a message thread (204);

identify one or more file attachments (124) attached to the one or more messages (118);

identify respective root files related to the one or more file attachments (124);

determine respective version levels of the one or more file attachments (124) based upon the root file related to the file attachment (124);

generate an aggregated attachment list (216) comprising respective list entries (218A-218C) for the root files identified; and

display the aggregated attachment list (216) in a window (202) of an email application (108).

11. The computer readable storage medium of claim 10, wherein the list entries respectively comprise one or more of a file-type icon, a file name, a file size, and an identification of an author of the file attachment related to the root files having a greatest version level, and an indication of a number of file attachments related to the root files.

12. The computer readable storage medium of claim 10, wherein identifying the one or more messages associated with the message thread comprises identifying one or more messages having a same conversation ID.

13. The computer readable storage medium of claim 10, wherein identifying the one or more messages associated with the message thread comprises identifying one or more messages having a same or similar subject.

14. The computer readable storage medium of claim 10, wherein identifying the respective root files related to the one or more file attachments comprises identifying file attachments having a same or similar file name as related to one root file.

15. The computer readable storage medium of claim 10, wherein determining respective version levels for the one or more file attachments based upon the root file related to the file attachment comprises comparing a respective modified date for the one or more file attachments related to the root file.

16. A system for displaying an aggregated attachment list (216) comprising:

a message store (116) containing a plurality of messages (118) associated with a message thread (204);

an email application (108) configured to execute on a workstation (102) and operable to access the message store (116) and display a window (202) comprising information regarding the plurality of messages (118); and

an attachment aggregation module (136) configured to execute within the email application (108) and operable to:

identify the plurality of messages (118) associated with the message thread (204),

identify one or more file attachments (124) attached to the plurality of messages (118),

identify respective root files related to the one or more file attachments (124),

generate the aggregated attachment list (216) to include respective list entries (218A-218C) for the root files identified, and

display the aggregated attachment list (216) in the window (202) of the email application (108).

17. The system of claim 16, wherein identifying the plurality of messages associated with the message thread comprises identifying the messages is the message store having a same conversation ID.

18. The system of claim 16, wherein identifying the respective root files related to the one or more file attachments comprises identifying file attachments having a same or similar file name as being related to a same root file.

19. The system of claim 16, wherein the message store is stored locally on the workstation.

20. The system of claim 16, wherein the message store is stored on a server computer remote from the workstation.